Linear Finite Transducers Towards a Public Key Cryptographic System

Ivone de Fátima da Cruz Amorim

Tese de Doutoramento apresentada à
Faculdade de Ciências da Universidade do Porto,
Ciência de Computadores

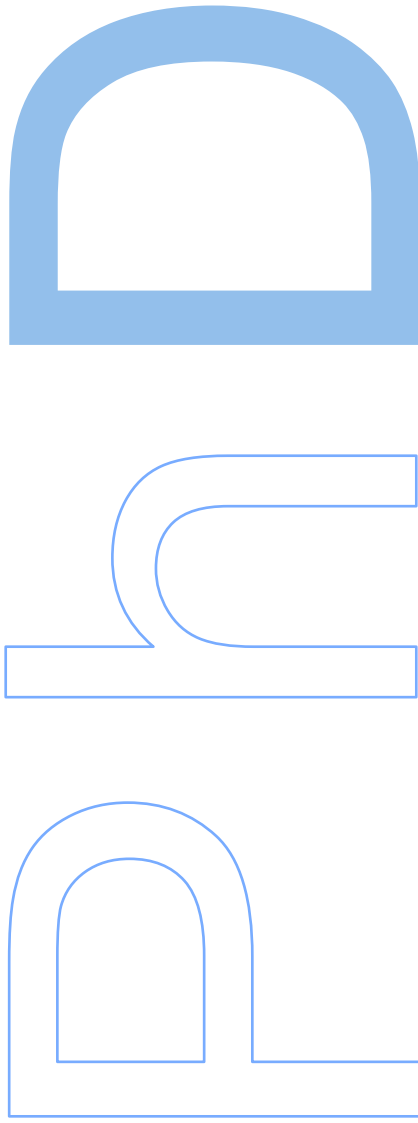
2016

Linear Finite Transducers Towards a Public Key Cryptographic System

Ivone de Fátima da Cruz Amorim
Doutoramento em Ciência de Computadores
Departamento de Ciência de Computadores
2016

Orientador
Rogério Ventura Lages dos Santos Reis, Professor Auxiliar,
Faculdade de Ciências da Universidade do Porto.

Coorientador
António José de Oliveira Machiavelo, Professor Auxiliar,
Faculdade de Ciências da Universidade do Porto.



To my father, who taught me the true meaning of courage.

Ao meu pai, por me ensinar o verdadeiro significado da palavra coragem.

Acknowledgments

I would like to take this opportunity to express my gratitude to a few people that supported me throughout the course of this project. First of all, I would like to acknowledge my supervisors, António Machiavelo and Rogério Reis, for their unconditional support from the beginning of this adventure. I thank them for the long hours they spent with me, which went far beyond what I could demand, for all the wise opinions they gave me about my work (and not only about the work), for all the questions they raised, which were fundamental for my growth as a researcher, for their (almost) infinite patience with my doubts and insecurities, and, finally, for the sense of humor that was always present in our meetings. I will always be grateful to them.

To Professor Renji Tao I thank the celerity with which he has always replied to my emails, and I also thank him for sending me a copy of documents that otherwise would be almost impossible to obtain.

I thank Professor Stavros Konstantidinis for his invitation to spend a month in Saint Mary's University as a visiting scholar. I also thank him for his kindness, hospitality and for all the scientific discussions I was able to have during my stay in Halifax.

To Nelma Moreira I thank for being always available when I needed her. I also thank her, and Rogério, for hosting me in the house they rented in Halifax, for the availability and care they always showed, and for all the exploring trips and conversations we had during my stay.

To Alexandra and Isabel for being always so efficient and helpful with all the bureau-

cratic questions, and for the good moments we shared during our "knitting meetings".

To my colleagues in general for their constant encouragement. To Cristina Lima for her prompt availability to proofread some parts of this thesis. A very special thanks goes to Eva Maia, with whom I shared much more than an office during my PhD. I thank her for all the conversations we had on the most diverse subjects, for all the opinions she gave me, always with different points of view, for the patience she showed to listen to my problems, even when she also needed support, and, mostly, for the moments we laughed together when we just wanted to cry.

To my siblings, Elisa, Fernando and Rui, for all the care over the years, for their encouragement, and for tolerating my bad mood in complicated moments. To my stepmother for the fundamental values she taught me. To my nieces and nephews, Beatriz, Bianca, Celia, Simão and Javier, I thank for all the moments we played together, which brought a lot of happiness to my life. To my sisters in law, Susete and Cristina, I thank for all the conversations and for always being so supportive.

I thank Paulo for the care and comprehension with which he always dealt with my absences, for listening to me and for the encouragement he gave me when I was questioning myself, for believing in my work, and for helping me to focus in what was important in the last phase of this journey.

Finally, I thank my father to whom I own the basis of my education. I thank him for always giving me the freedom to choose my way, for stimulating my critical spirit, and for showing me, through his own example, that we can make our dreams come true. Above all, I thank him for making the well-being of our family his priority, when we most needed him.

Regarding financial support, I thank Fundação para a Ciência e Tecnologia for the PhD grant [SFRH/BD/84901/2012], and to Centro de Matemática da Universidade do Porto for funding all my conference participations.

Agradecimentos

Aproveito esta oportunidade para fazer um pequeno agradecimento a algumas pessoas que me apoiaram ao longo deste trabalho. Em primeiro lugar, quero agradecer aos meus orientadores, António Machiavelo e Rogério Reis, pelo apoio incondicional desde o início desta aventura. Agradeço pelas longas horas que me dispensaram, que foram muito além do que eu poderia exigir, por todas as opiniões sábias que deram sobre o meu trabalho (e não só), por todas as questões que colocaram, que foram fundamentais no meu crescimento enquanto investigadora, pela sua (quase) infinita paciência para as minhas dúvidas e inseguranças e, finalmente, pelo sentido de humor que esteve sempre presente nas nossas reuniões.

Ao Professor Renji Tao agradeço a rapidez com que sempre respondeu aos meus emails e por tão prontamente me ter disponibilizado documentos que de outra forma eu dificilmente conseguiria obter.

Agradeço ao Professor Stavros Konstantinidis pelo convite para passar um período na *Saint Mary's University* na qualidade de *visiting scholar*. Agradeço, ainda, pela sua simpatia, hospitalidade e por todas as discussões científicas em que pude participar durante a minha estadia em Halifax.

À Nelma Moreira agradeço toda a disponibilidade que sempre demonstrou nas mais diversas situações em que precisei da sua ajuda. Agradeço-lhe, ainda, tal como agradeço ao Rogério, por me terem acolhido na casa que alugaram em Halifax, pela disponibilidade e preocupação que sempre demonstraram e por todos os passeios e conversas que tivemos durante a minha estadia.

Agradeço à Alexandra e à Isabel por tão eficientemente me terem ajudado na resolução de todas as questões burocráticas que foram surgindo e por todos os bons momentos que partilhamos durante as nossas "reuniões do tricô".

Agradeço a todos os meus colegas que, de alguma forma, me incentivaram. À Cristina Lima por se ter disponibilizado tão prontamente a ler partes desta tese e por ter estado sempre disponível para me ouvir. Deixo um agradecimento muito especial à Eva Maia, com quem partilhei muito mais do que um gabinete durante o meu doutoramento. Agradeço-lhe pelas nossas conversas sobre os mais diversos assuntos, por todas as opiniões que me deu com pontos de vista sempre diferentes, pela paciência com que ouviu os meus desabafos mesmo quando ela também precisava de apoio e, principalmente, por todos os momentos em que nos rimos, quando só nos apetecia chorar.

Agradeço aos meus irmãos, Elisa, Fernando e Rui, por todo o carinho que me deram ao longo da minha vida, por me incentivarem e por tolerarem o meu mau humor em momentos mais complicados. À minha madrasta, agradeço pelos valores fundamentais que me transmitiu. Aos meus sobrinhos, Simão, Beatriz, Bianca, Celia e Javier, agradeço por todas as travessuras e momentos de brincadeira que partilhamos, momentos esses que tornaram a minha vida muito mais feliz. Às minhas cunhadas, Susete e Cristina, agradeço por todas as conversas que tivemos e por sempre me terem apoiado.

Agradeço ao Paulo pelo carinho e pela compreensão com que sempre lidou com as minhas ausências. Por me ter ouvido e incentivado nas imensas vezes em que duvidei de mim. Por ter acreditado no meu trabalho e por me ter ajudado a focar naquilo que era importante na fase final deste percurso.

Por fim, agradeço ao meu pai, a quem devo a base da minha educação. Agradeço-lhe por sempre me ter dado a liberdade de escolher o meu caminho, por ter estimulado o meu espírito crítico e por me ter mostrado, através do seu próprio exemplo, que é possível concretizarmos os nossos sonhos. Acima de tudo, agradeço-lhe por ter feito

do bem-estar da nossa família a sua prioridade quando mais precisamos.

No que diz respeito ao suporte financeiro, agradeço à Fundação para a Ciência e Tecnologia pela bolsa de doutoramento [SFRH/BD/84901/2012] e ao Centro de Matemática da Universidade do Porto por financiar todas as despesas inerentes às minhas deslocações às várias conferências.

Abstract

Cryptography faces a set of new challenges. The rapid advance in computing power and technology, as well as the possibility of quantum computing becoming a reality, are real threats to the security offered by classical cryptographic systems. New cryptographic systems, relying in different assumptions, are needed.

Cryptographic systems based on finite transducers are an exciting possible solution to these new challenges. First, their security does not rely on complexity assumptions related to number theory problems (as classical systems do), it relies on the apparent difficulties of inversion of non-linear finite transducers and of factoring matrix polynomials over $\mathbb{F}_q$. Secondly, they offer relatively small key sizes as well as linear encryption and decryption times complexity.

The techniques used in these systems depend heavily on the results of invertibility of linear finite transducers (LFTs). In this thesis we give a complete characterisation of LFTs, while discussing their invertibility. A wide variety of examples are presented in order to illustrate the concepts and techniques proposed.

The main original contributions of this work are the following.

- An equivalence test for LFTs.
- A canonical representation for LFTs, and an algorithm to compute such a representation.
- Methods to compute the number and size of equivalence classes of LFTs defined

over $\mathbb{F}_q$, and an algorithm to enumerate all the equivalent LFTs with the same number of states.

- The implementation of an algorithm that employees a known condition, due to Zongduo and Dingfeng, to check τ-injectivity of LFTs.
- Methods to estimate the number and percentage of τ-injective equivalence classes ($\tau \in \mathbb{N}_0$), by uniform random generation of LFTs, and implementations of these methods in `Python` using some `Sage` modules to deal with matrices.
- An experimental study using these implementations.
- An extension of the concept of LFT with memory, called PILT, and a necessary and sufficient condition for the injectivity of these transducers.
- An algorithm to invert PILTs, which, since LFTs with memory are PILTs, allows to find left inverses of invertible LFTs with memory.

Resumo

A Criptografia enfrenta um conjunto de novos desafios. A rápida evolução da tecnologia e do poder computacional, assim como a possibilidade da computação quântica se tornar uma realidade, são ameaças sérias à segurança oferecida pelos sistemas criptográficos clássicos. São necessários novos sistemas criptográficos que assentem em diferentes pressupostos de complexidade.

Os sistemas criptográficos baseados em transdutores finitos são uma possível solução para estes novos desafios. Em primeiro lugar, a sua segurança não assenta em pressupostos de complexidade relacionados com problemas de teoria de números (tal como os sistemas clássicos), mas sim na dificuldade da inversão de transdutores finitos não lineares e na dificuldade da factorização de matrizes polinomiais. Por outro lado, os tamanhos da chave exigidos são relativamente pequenos e os tempos de cifra e decifração são lineares.

As técnicas usadas nestes sistemas dependem fortemente dos resultados existentes sobre a invertibilidade de transdutores finitos lineares (TFLs). Nesta tese dá-se uma caracterização completa destes transdutores e, ao mesmo tempo, discute-se a sua invertibilidade. Além disso, também é apresentada uma grande variedade de exemplos que permitem ilustrar os conceitos e técnicas aqui propostos.

As principais contribuições originais deste trabalho são as seguintes.

- Um teste que permite verificar a equivalência de TFLs.
- Uma representação canónica para TFLs e um algoritmo para determinar essa

representação.

- Métodos para calcular o número e o tamanho das classes de equivalência de TFLs definidos sobre $\mathbb{F}_q$ e um algoritmo que permite enumerar todos os TFLs equivalentes que têm o mesmo número de estados.

- A implementação de um algoritmo que aplica uma condição já conhecida para verificar se um TFL é τ-injectivo.

- Métodos para estimar o número e a percentagem de classes de equivalência τ-injectivas, usando geração aleatória uniforme de TFLs, e implementações destes métodos em `Python` usando alguns módulos do `Sage` para trabalhar com matrizes.

- Um estudo experimental usando estas implementações.

- Uma extensão do conceito de TFL com memória, chamada PILT, e uma condição necessária e suficiente para a injectividade destes transdutores.

- Um algoritmo para inverter PILTs que, uma vez que os TFLs com memória são PILTs, permite encontrar um inverso à esquerda de qualquer TFL com memória que seja injectivo.

Resumé

La Cryptographie est aujourd'hui devant des nouveaux défis. L'avance rapide de la puissance de calcul des ordinateurs et de la technologie, ainsi que la possibilité des ordinateurs quantiques devient une réalité, sont de sérieux menaces à la sécurité offerte par des systèmes cryptographiques classiques. Des nouveaux systèmes cryptographiques en se fondant dans différentes hypothèses de complexité sont donc nécessaires.

Les systèmes cryptographiques édifiés sur les transducteurs finis constitue une solution prometteuse à ces nouveaux défis. Tout d'abord, leur sécurité ne repose pas dans les hypothèses de la complexité des problèmes liés à la théorie des nombres (comme pour les systèmes classiques), elle repose sur les apparentes difficultés de l'inversion des automates finis non linéaires et de la factorisation des polynômes matriciels sur $\mathbb{F}_q$. Deuxièmement, ils offrent des clés à tailles relativement petites, ainsi qu'une chiffrage et le déchiffrage à temps linéaire.

Les techniques utilisées dans ces systèmes dépendent fortement des résultats de l'inversibilité de transducteurs finis linéaires (TFLs). Dans cette thèse, on donne une caractérisation complète de TFLs et on discute de leur inversibilité. Des différent exemples sont données pour illustrer les concepts et les techniques proposées.

Les principales contributions originales de ce travail sont les suivants :

- Un algoritme pour tester l'équivalence de TFLs.
- Une représentation canonique pour TFL et un algorithme pour calculer cette représentation.

- Méthodes pour calculer le nombre d'éléments et la taille des classes d'équivalence de transducteurs finis définies sur $\mathbb{F}_q$ qui sont τ-injective ($\tau \in \mathbb{N}_0$), et un algorithme pour énumérer tous les TFLs équivalentes qui ont le même nombre d'états.

- La implémentation d'un algorithme en utilisant une condition de Zongduo et Dingfeng pour vérifier la τ-injectivité de TFLs.

- Méthodes pour estimer le nombre et le pourcentage de classes d'équivalence qui sont τ-injective, pour génération aléatoire uniforme de TFLs, et des implémentations de ces méthodes en `Python` utilisant certains modules de `Sage` pour le traitement des matrices.

- Une étude expérimentale utilisant ces implémentations.

- Une extension de la notion de TFL avec mémoire, que nous avons appelé PILT, et une condition nécessaire et suffisante pour l'injectivité de ces transducteurs.

- Un algorithme pour inverser PILTs, qui, une fois que les TFLs avec mémoire sont PILTs, permet de trouver inverses gauche des TFLs avec mémoire qui sont inversibles.

Contents

Acknowledgments **vii**

Agradecimentos **ix**

Abstract **xiii**

Resumo **xv**

Resumé **xvii**

List of Tables **xxiii**

List of Figures **xxv**

List of Algorithms **xxvii**

1 Introduction **1**

1.1 Structure of this Dissertation 5

2 Mathematical Prerequisites **7**

2.1 Relations and Funtions 7

2.2 Groups, Rings, PIDs, and Fields 9
2.3 Modules and Vector Spaces 14
2.4 Matrices and Smith Normal Form 16
2.5 Cayley-Hamilton Theorem and Some Implications 25
2.6 Linear Maps 27

3 Linear Finite Transducers **31**

3.1 Preliminaries on Finite Transducers 31
3.1.1 Concepts on Invertibility 43
3.1.2 Finite Transducers with Memory 48
3.2 The Notion of Linear Finite Transducer 51
3.3 Equivalence of States and of LFTs 54
3.4 Minimisation 61

4 Size and Number of Equivalence Classes of LFTs **65**

4.1 Canonical Linear Finite Transducers 65
4.2 Size of Equivalence Classes 69
4.3 Number of Equivalence Classes 76

5 Equivalence Classes of Injective LFTs **81**

5.1 Injectivity of LFTs 81
5.2 Number of Injective Equivalence Classes 88
5.3 Percentage of Injective Equivalence Classes 92

5.4 Experimental Results 95

6 Inverses of LFTs with Memory **101**

6.1 Linear Finite Transducers with Memory 101

6.2 Injectivity of LFTs with Memory 104

6.3 Post-Initial Linear Transducers 108

7 Conclusion **125**

A Tables of Experimental Results **129**

B Change of Variables in Summations **131**

Index **136**

List of Tables

4.1 Enumeration of transducers in $\hat{S}_2$ and $\hat{S}_X$ of Example 4.9. 77

5.1 Approximated values for the number of injective equivalence classes when $m = 5$ and $\tau = 10$. 95

6.1 Coefficients of Θ. 109

A.1 Estimates of the percentage of τ-injective equivalence classes for $\ell = 2$ and $m = 5$. 129

A.2 Estimates of the percentage of τ-injective equivalence classes for $\ell = 3$ and $m = 5$. 129

A.3 Estimates of the percentage of τ-injective equivalence classes for $\ell = 4$ and $m = 5$. 130

A.4 Estimates of the percentage of τ-injective equivalence classes for $\ell = 5$ and $m = 5$. 130

A.5 Estimates of the percentage of τ-injective equivalence classes for $\ell = 8$ and $m = 8$. 130

List of Figures

5.1 Variation on the percentage of τ-injective equivalence classes for $\ell = 2$, $m = 5$, and several values of n and τ (from two different perspectives). 96

5.2 Variation on the percentage of τ-injective equivalence classes for $m = 5$ and several values of ℓ, n and τ. 97

5.3 Variation on the percentage of τ-injective equivalence classes for $m = 5$ and several values of ℓ, n and τ (from a different perspective than that from Figure 5.2). 98

5.4 Variation on the percentage of τ-injective equivalence classes for $\ell = 8$, $m = 8$, and several values of n and τ (from two different perspectives). 99

List of Algorithms

5.1 Testing the injectivity. 87

5.2 Determining the size of equivalence classes. 90

5.3 Estimating the number of non-equivalent LFTs. 91

5.4 Auxiliary functions. 92

5.5 Counting the number of canonical LFTs. 93

5.6 Estimating the percentage of injective equivalence classes. 94

Chapter 1

Introduction

The concept of Public Key Cryptography (PKC) was introduced by Diffie, Hellman and Merkle in 1976. In 1978, Rivest, Shamir and Adleman presented the first public key cryptosystem, called RSA [Dif88]. The RSA system, and most of the public key cryptosystems created in the following years, are based on complexity assumptions related to number theory problems, namely the factorisation of integers and the discrete logarithm problem. This dependence on a very small set of problems makes such cryptosystems somewhat vulnerable. Also, improvements in algorithms to solve these problems have led to the need of increasing the size of the keys, which implies higher computational costs. Moreover, the past few years have witnessed an astonishing increase on the diversity of small computing devices allowing to implement almost every kind of digital service that, up to now, were only possible on computers. These small devices are very attractive, and are now affordable by almost everyone. However, they have very limited resources, which requires new cryptographic solutions that should be both secure and extremely fast.

In a series of papers [TC85, TCC97, TC97, TC99], Renji Tao introduced a family of cryptosystems based on finite transducers, named FAPKCs (which stands for Finite Automata Public Key Cryptosystems), which seems to be a good alternative to the classical ones. First, the security of these systems does not rely on complexity as-

sumptions related to number theory problems (as classical systems do), rather relying on the difficulty of inverting non-linear finite transducers and of factoring matrix polynomials over $\mathbb{F}_q$ [Tao09]. The complexity of these problems is not known, apart from the trivial fact that they are both NP-problems, exactly like the integer factoring problem that is the basis of RSA. Secondly, they offer relatively small key sizes as well as fast encryption and decryption [TC97, Abu11]. This makes them computationally attractive, and thus suitable for application on devices with very limited computational resources, such as satellites, cellular phones, sensor networks, and smart cards [TC97]. Besides, the FAPKC schemes are stream ciphers that can be used for encryption and signature [Tao09].

The first FAPKC system was proposed in 1985 by Tao and Chen in a paper (in Chinese) and was named FAPKC0. An English description of it was presented in a later work of the same authors [TC86]. Roughly speaking, in this system, the private key consists of two injective transducers with memory, where one is a linear finite transducer (LFT), M, and the other is a non-linear finite transducer (non-LFT), N, whose left inverses can be easily computed. The public key is the result of applying a special product for transducers, $\mathcal{C}$, to the original pair, thus obtaining a non-LFT, denoted by $\mathcal{C}(M, N)$. The crucial point is that it is easy to obtain an inverse of $\mathcal{C}(M, N)$ from the inverses of its factors, M^{-1} and N^{-1}, while it is believed to be hard to find that inverse without knowing those factors. On the other hand, the factorisation of a transducer seems to be hard by itself [ZDL98].

The system FAPKC0 was derived mainly from the results about invertibility on LFTs presented by Tao in 1973 [Tao73], which was the first relevant work on invertibility theory of finite transducers with applications to Cryptography. In 1986, Tao and Chen published two variants of that cryptosystem, named FAPKC1 and FAPKC2 [TC86], but with no further advances on the invertibility theory of finite transducers. In 1992, the methods used to study the invertibility of LFTs were applied to quasi-linear finite transducers over finite fields (as defined by Tao [Tao09]). And, in 1995, they were generalised to construct pairs of transducers in which one is a left inverse

of the other. This new development on the invertibility theory of finite transducers gave rise to two new cryptographic schemes: FAPKC3 and FAPKC4, presented by Tao *et al.* [TCC97] and by Tao and Chen [TC97], respectively. Meanwhile, some other schemes of Public Key Cryptography based on finite transducers were developed (the system FAPKC93 was presented in a PhD thesis written in Chinese, and a variant of FAPKC2 was put forward by Bao and Igarashi [BI95]). All of these systems are similar in structure, their main difference being the choice of the transducers for the private key. For example, while in the FAPKC0 system M is linear and N is non-linear, in the system FAPKC3 the transducers M and N are both non-linear. The systems FAPKC0, FAPKC1, FAPKC93 and the variant of FAPKC2, were proved to be insecure [Tao95a, Tao95b, TC97]. The systems FAPKC2, FAPKC3 and FAPKC4 have not yet been adequately evaluated.

Although some of the FAPKC schemes were already shown to be insecure, the promise of a new system of PKC relying on different complexity assumptions makes these systems worth exploring. However, the uninspiring and arid language used in Tao's works seems to have condemned these systems to oblivion. Moreover, the study of finite transducers and their invertibility is spread over a series of papers that sometimes do not contain proofs, or refer to papers that are written in Chinese and/or are not easily available to the English reader. Also, there is an almost total lack of examples, making it difficult to understand the underlying theory. From all this, it is clear that, on the one hand, there is a need for a clarification and consolidation of the work already done in this subject and, on the other hand, it is necessary to do a serious study of these systems and their application. This thesis is a starting point in that direction.

In this work, we give an unified presentation of the known results, as far as we can establish, on general linear finite transducers as well as on linear transducers with memory. We also simplify the language used, by introducing a more classical point of view.

As our first contribution we present a new equivalence test for LFTs which is of paramount importance in the following work. We then give a complete characterisation

of these transducers, by introducing a notion of canonical LFT and by studying the number and size of LFT's equivalence classes. An algorithm to enumerate the LFTs in the same equivalence class is also provided.

We then show how to estimate the number and percentage of non-equivalent LFTs that are τ-injective ($\tau \in \mathbb{N}_0$), by uniform random generation of LFTs. This number is fundamental to evaluate the key space of cryptographic systems that use this kind of transducers, and their percentage is crucial to conclude if uniform random generation of non-equivalent LFTs is a feasible option to generate cryptographic keys. As far as we know, no similar study has ever been conducted. All the algorithms presented were implemented in `Python` using some `Sage` [Dev15] modules to deal with matrices. Several experiments were carried out and the results obtained are also given, which by themselves constitute an important step towards the evaluation of these systems.

Finally, we address the invertibility problem in LFTs with memory. Inverting transducers of this kind is fundamental in the key generation process of FAPKCs that use LFTs, since one needs to define both an invertible LFT with memory and a corresponding left inverse. Moreover, new techniques to invert injective finite transducers may allow to study the vulnerability of the existent cryptographic systems from novel points of view. Despite the works done on the invertibility of LFTs [Tao73, Tao88, ZD96, ZDL98, HZ99], none of them presents an algorithm to invert LFTs with memory. Thus, in this work, we introduce the notion of post-initial linear transducer (PILT), which is an extension of the notion of LFT with memory, and give explicitly an algorithm to invert this kind of transducers.

We also present, throughout this work, a wide variety of examples to illustrate the concepts and techniques proposed.

1.1 Structure of this Dissertation

We start by reviewing, in Chapter 2, several concepts and some results from different areas of mathematics that will be used throughout this work. We also introduce some convenient notation.

Preliminary notions and results of general finite transducers are given in Chapter 3, including the concepts of injectivity and invertibility that are considered in this work. Also, in this chapter, we give the definition of LFT, present some already known results, and give our new method to check LFT's equivalence. At the end, we discuss the minimisation problem of these transducers.

In Chapter 4, we give our notion of canonical LFT and prove that each equivalence class has exactly one of these transducers. We also show how to construct the canonical LFT equivalent to an LFT given in its matricial form. Then, by using the new equivalence test for LFTs presented in Chapter 3, we enumerate and count the equivalent transducers with the same size. From this, we derive a recurrence relation that counts the number of equivalence classes, *i.e.*, the number of non-equivalent LFTs.

Chapter 5 is devoted to the statistical study on the number and percentage of τ-injective equivalence classes. We start by reviewing some results on the invertibility of LFTs and by giving an algorithm to test if an LFT is injective with some delay $\tau \in \mathbb{N}_0$. Then, we show how to estimate the number of τ-injective equivalence classes, using the results of the previous chapter about the size of equivalence classes. After that, we deal with the problem of computing the percentage of τ-injective equivalence classes, using the estimate for the number of those classes and the fact that each equivalence class has exactly one canonical LFT. We end this chapter with a presentation and discussion of our experimental results.

The invertibility problem in LFTs with memory is dealt with in Chapter 6. We first discuss the form of the structural matrices for LFTs with memory, and then we study how that form allows to simplify the method presented in the previous chapter to check

injectivity of LFTs. The notion of PILT is then introduced as well as the method we propose to compute left inverses of invertible PILTs. Since an LFT with memory is also a PILT, this method allows to invert any injective LFT with memory.

Finally, in Chapter 7, we summarise our contributions and discuss some future research directions.

Some of the results here included were previously presented in conferences of the area or published in scientific journals [AMR14a, AMR14c, AMR15, AMR12, AMR14b].

Chapter 2

Mathematical Prerequisites

2.1 Relations and Funtions

Let A and B be two sets. A *relation* $\sim$ from A to B is a subset of the cartesian product $A \times B$. We write $a \sim b$ to denote that (a, b) is in the relation $\sim$. If (a, b) is not in the relation $\sim$, we write $a \not\sim b$. When $A = B$, $\sim$ is also called *a binary relation on* A.

A binary relation $\sim$ on a set A is said to be an *equivalence relation* if and only if the following conditions hold:

- $\sim$ is reflexive, *i.e.*, $a \sim a$, for all a in A;
- $\sim$ is symmetric, *i.e.*, $a \sim b$ if and only if $b \sim a$, for all a, b in A;
- $\sim$ is transitive, *i.e.*, if $a \sim b$ and $b \sim c$, then $a \sim c$, for all a, b, c in A.

Let $\sim$ be an equivalence relation on A. For any $a \in A$, the set $[a]_\sim = \{b \in A \mid a \sim b\}$ is called the *equivalence class* containing a, while the set of all equivalence classes, $A/\sim = \{[a]_\sim \mid a \in A\}$, is called the *quotient of* A *by* $\sim$.

The *restriction of a binary relation* on a set A to a subset S is the set of all pairs (a, b) in the relation for which a and b are in S. If a relation is an equivalence relation, its

restrictions are too.

Given a positive integer n, an example of an equivalence relation is the *congruence modulo n* relation on the set of integers, $\mathbb{Z}$. For a positive integer n, one defines this relation on $\mathbb{Z}$ as follows. Two integers a and b are said to be *congruent modulo n*, written:

$$a \equiv_n b \quad \text{or} \quad a \equiv b \pmod{n},$$

if their difference $a-b$ is a multiple of n. It is easy to verify that this is an equivalence relation on the integers. The number n is called the *modulus*. An equivalence class consists of those integers which have the same remainder on division by n. The set of *integers modulo n*, which is denoted by $\mathbb{Z}_n$, is the set of all congruence classes of the integers for the modulus n.

Example 2.1. *Take $n = 2$. Then, for example,*

$$5 \equiv 3 \equiv 1 \pmod{2} \quad \textit{and} \quad [1]_\sim = \{2j+1 \mid j \in \mathbb{Z}\}.$$

A relation from a set A to a set B is called a *function*, *map* or *mapping*, if each element of A is related to exactly one element in B. A function f from A to B is denoted by $f : A \to B$, and for all a in A, $f(a)$ denotes the element in B which is related to a, which is usually called the *image of a under f*.

A function $f : A \to B$ is called *injective*, or a *one-to-one* function, if it satisfies the following condition:

$$\forall\, a, a' \in A,\ f(a) = f(a') \Rightarrow a = a',$$

and is called *surjective* if the following condition holds:

$$\forall\, b \in B,\ \exists\, a \in A,\ f(a) = b.$$

If a function is both injective and surjective, then it is called *bijective* or a *bijection*.

2.2 Groups, Rings, PIDs, and Fields

Let A be a set and n a natural number. A *n-ary operation* on A is a mapping from A^n to A. We call $\diamond : A^2 \to A$ a *binary operation*, which only means that if (a, b) is an ordered pair of elements of A, then $a \diamond b$ is a unique element of A.

A *group* is an ordered pair $(G, \diamond)$, where G is a non-empty set and $\diamond$ is a binary operation on G (called the *group operation*), satisfying the following properties:

- the operation $\diamond$ is associative, that is, $x \diamond (y \diamond z) = (x \diamond y) \diamond z$, for all $x, y, z \in G$;
- there is an element $e \in G$ such that $x \diamond e = e \diamond x = x$, for all x in G. Such an element is unique and is called the *identity element*;
- if x is in G, then there is an element y in G such that $x \diamond y = y \diamond x = e$, where e is the identity element. That element y is called the *inverse* of x.

We say that a group is *denoted additively* (*multiplicatively*) or is an *additive* (*multiplicative*) *group* when:

- the group operation is denoted by $+$ $(\cdot)$;
- the identity element is denoted by 0 (1);
- the inverse of an element x is denoted by $-x$ (x^{-1}),

respectively. If the group operation is commutative, *i.e.*, $x \diamond y = y \diamond x$ for all x, y in G, then G is called an *Abelian group* or *commutative group*.

There are some very familiar examples of Abelian groups under addition, namely the integers $\mathbb{Z}$, the rationals $\mathbb{Q}$, the real numbers $\mathbb{R}$, and $\mathbb{Z}_n$, for $n \in \mathbb{N}$. Notice that $\mathbb{N}$ denotes de set of natural numbers, *i.e.*, $\mathbb{N} = \{1, 2, 3, \ldots\}$.

A *ring* is an ordered triple $(R, +, \cdot)$, where R is a non-empty set, $+$ is a binary operation on R called *addition*, and $\cdot$ is also a binary operation on R called *multiplication*, which obey the following rules:

- $(R, +)$ is an Abelian group (the additive identity is denoted by 0);
- the multiplicative operation is associative, that is, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, for all x, y, x in R;
- there is an element 1 in R such that $1 \cdot x = x \cdot 1 = x$, for all x in R. 1 is called the *multiplicative identity*;
- the multiplication is left distributive with respect to addition, that is, $x \cdot (y+z) = x \cdot y + x \cdot z$, for all x, y, z in R;
- the multiplication is right distributive with respect to addition, *i.e.*, $(x+y) \cdot z = x \cdot z + y \cdot z$, for all x, y, z in S.

A simple example of a ring is the set of integers with the usual operations of addition and multiplication.

Let R be a ring with multiplicative identity 1. An element r in R is said to be *multiplicatively invertible* or just *invertible* if and only if there is an element s in R such that $r \cdot s = s \cdot r = 1$, and s is called the *multiplicative inverse* of r or just the *inverse* of r. An invertible element in R is called a *unit* and the set of units of R is represented by R^*. Let $a, b \in R$. We say that a *divides* b, and write $a \mid b$, if there is $q \in R$ such that $b = aq$, where aq abbreviates $a \cdot q$. The definition of congruence modulo n relation on the set of integers, presented in page 8, can be generalised to elements of a ring. Thus, we say that two elements, a, b, in a ring, R, are *congruent modulo* $n \in R$ if $n \mid (a - b)$.

The *ring of polynomials* in the variable x with coefficients in a ring R is denoted by $R[x]$ and is formed by the set of polynomials in x and the usual operations of polynomial addition and multiplication. A polynomial in $R[x]$ is therefore an expression of the form

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n,$$

for some $n \in \mathbb{N}_0$, and where $a_i \in R$, for all $0 \leq i \leq n$. Recall that if $p(x)$ is a non-zero element of $R[x]$, and n is the largest non-negative integer such that x^n has a non-zero

coefficient in p, then one says that p has *degree* n or that p is a *polynomial* of order n, and denote this by $\deg(p) = n$. In this context, $P_n(R[x])$ stands for the set of polynomials in $R[x]$ that have degree less than n. If $n = 0$ the polynomial is said to be *constant*, while if $n = 1$ is said to be *linear*. A *monic polynomial* is a polynomial in which the coefficient of the highest order term is 1. The invertible elements in $R[x]$ are just the constant polynomials a_0 with a_0 invertible in R.

Another important example of a ring, for this work, is the ring of formal power series over an arbitrary ring. Roughly speaking, the *formal power series* are a generalisation of polynomials as formal objects, where the number of terms is allowed to be infinite, that is, a formal power series over a ring R is an expression of the form

$$f(x) = \sum_{i \geq 0} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots ,$$

where $a_i \in R$, for all $i \in \mathbb{N}_0$. Addition and multiplication are defined just as for the ring of polynomials $R[x]$:

$$\sum_{i \geq 0} a_i x^i + \sum_{i \geq 0} b_i x^i = \sum_{i \geq 0} (a_i + b_i) x^i,$$

$$\left(\sum_{i \geq 0} a_i x^i \right) \left(\sum_{j \geq 0} b_j x^j \right) = \sum_{k \geq 0} c_k x^k, \text{ where } c_k = \sum_{i+j=k} a_i \cdot b_j.$$

The *ring of formal power series* in the variable x with coefficients in the ring R is denoted by $R[[x]]$, and is formed by the set of power series in x with the addition and multiplication operations as defined above. The invertible elements in $R[[x]]$ are the power series whose constant term is invertible in R.

When a ring multiplicative operation is commutative, the ring is said to be a *commutative ring*. For example, the rings $\mathbb{Z}$, $\mathbb{Z}[x]$ and $\mathbb{Z}[[x]]$ are all commutative.

An *ideal* is a subset I of a ring R with the following properties:

- $I \neq \emptyset$;

- the ideal is closed under addition, *i.e.*, $r + s \in I$, for all r, s in I;
- the product of an element of the ideal and an element of the ring is an element of the ideal, *i.e.*, $ri \in I$ and $ir \in I$, for all r in R, and for all i in I.

The set of even integers, denoted by $2\mathbb{Z}$, is an ideal of the ring $\mathbb{Z}$. This is easy to check because $0 \in 2\mathbb{Z}$, the sum of any two even integers is even, and the product of any even integer by an integer is also even. The ideal $2\mathbb{Z}$ is also an example of what is called an ideal generated by a single element. Let $n \in \mathbb{N}$ and $S = \{s_1, \ldots, s_n\}$ be a subset of R. The *ideal generated by* S is the subset

$$\left\{ \sum_{i=1}^{n} r_i s_i \mid r_i \in R \right\}.$$

A *Principal Ideal Domain* (PID) is a non-zero commutative ring in which every ideal can be generated by a single element. Principal ideal domains are mathematical objects that behave somewhat like the integers with respect to divisibility. For example, like the integers, any element of a PID has a unique decomposition into prime elements, that is, a PID is a unique factorisation domain. The ring of integers $\mathbb{Z}$ is a PID. On the other hand, the ring of polynomials $\mathbb{Z}[x]$ is not a PID because, for example, the ideal generated by 2 and x, $\{2r_1 + xr_2 \mid r_1, r_2 \in \mathbb{Z}[x]\}$, is an example of an ideal in $\mathbb{Z}[x]$ that is not generated by a single polynomial in $\mathbb{Z}[x]$.

Given a ring R in which not all non-zero elements are multiplicatively invertible, we can extend that ring in such a way that more of its elements become invertible, by introducing "fractions".

If R is a ring, one says that a subset S of R is a *multiplicatively closed set* if and only if the following two conditions are true:

1. $1 \in S$;
2. $\forall x, y \in S,\ xy \in S$.

Let S be the multiplicative closed subset of R formed by the elements that we would like to become invertible. Consider the equivalence relation on the set $R \times S$ defined by

$$(r_1, s_1) \sim (r_2, s_2) \iff r_1 s_2 = r_2 s_1,$$

and denote the equivalence class of a pair $(r, s) \in R \times S$ by $\frac{r}{s}$. Then, the *localisation of R with respect to S*, denoted by R_S, is the ring formed by the set

$$\left\{ \frac{r}{s} \;\middle|\; r \in R, s \in S \right\}$$

together with the following operations of addition and multiplication:

$$\frac{r_1}{s_1} + \frac{r_2}{s_2} = \frac{r_1 s_2 + r_2 s_1}{s_1 s_2} \quad \text{and} \quad \frac{r_1}{s_1} \times \frac{r_2}{s_2} = \frac{r_1 r_2}{s_1 s_2}.$$

The localisation ring of R with respect to the set of all non-zero elements which are not multiplicatively invertible, *i.e.*, with respect to $S = R \setminus (R^* \cup \{0\})$, is referred to as the *ring of fractions* of R. A simple example of a localisation ring construction is the way that the set of rational numbers, $\mathbb{Q}$, is constructed from the integers, $\mathbb{Z}$.

A *field* is a commutative ring that has multiplicative inverses for all non-zero elements.

The set of real numbers, together with the usual operations of addition and multiplication, is a field. The commutative ring $\mathbb{R}[x]$ is not a field because not all non-zero polynomials in $\mathbb{R}[x]$ have multiplicative inverses (only the non-zero constant polynomials are invertible).

If $\mathbb{F}$ is a field with a finite number of elements, then one says that $\mathbb{F}$ is a *finite field* or a *Galois field*. The simplest examples of finite fields are the prime fields: given a prime number p, the prime field $GF(p)$ or $\mathbb{F}_p$ is the set of integers modulo p, previously denoted by $\mathbb{Z}_p$. The elements of a prime field may be represented by integers in the range $0, 1, \ldots, p-1$. For example,

$$\mathbb{F}_2 = \{0, 1\}.$$

2.3 Modules and Vector Spaces

Let R be a ring and 1 its multiplicative identity. A *right R-module*, M, consists of an Abelian group $(M, +)$ and an operation $\bullet : M \times R \to M$ such that for all $r, s \in R$ and $x, y \in M$, we have:

- $(x + y) \bullet r = x \bullet r + y \bullet r$
- $x \bullet (r + s) = x \bullet r + x \bullet s$
- $x \bullet (rs) = (x \bullet r) \bullet s$
- $x \bullet 1 = x$.

The operation of the ring on M is called *scalar multiplication*, and is usually written by juxtaposition, *i.e.*, xr for $r \in R$ and $x \in M$. However, in the definition above, it is denoted as $x \bullet r$ to distinguish it from the ring multiplication operation, which is denoted by juxtaposition. A *left R-module M* is defined similarly, except that the ring acts on the left, *i.e.*, scalar multiplication takes the form $\bullet : R \times M \to M$, and the above axioms are written with scalars r and s on the left of x and y.

If R is commutative, then left R-modules are the same as right R-modules and are simply called *R-modules*.

For example, if R is a commutative ring and $n \in \mathbb{N}$, then R^n is both a left and a right R-module if we use the component-wise operations:

$$(a_1, a_2, \ldots, a_n) + (b_1, b_2, \ldots, b_n) = (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n),$$

and

$$\alpha(a_1, a_2, \ldots, a_n) = (\alpha a_1, \alpha a_2, \ldots, \alpha a_n),$$

for all $(a_1, a_2, \ldots, a_n)$, $(b_1, b_2, \ldots, b_n) \in R^n$, and for all $\alpha \in R$.

Let $\mathbb{F}$ be a field. Then an $\mathbb{F}$-module is called a *vector space* over $\mathbb{F}$.

Example 2.2. *If $R = \mathbb{F}[[x]]$, where $\mathbb{F}$ is a field and x an indeterminate, then $F[[x]]^n$ is an R-module, for $n \in \mathbb{N}$.*

Example 2.3. *Let $n \in \mathbb{N}$. The set $\mathbb{F}_2^n$ with the component-wise operations of addition and scalar multiplication, as defined above, is a vector space over the field $\mathbb{F}_2$ which is denoted simply by $\mathbb{F}_2^n$.*

Let V be a vector space over a field $\mathbb{F}$. A non-empty subset U of V is said to be a *subspace* of V, if U is itself a vector space over $\mathbb{F}$ with the same operations as V.

Let V be a vector space over an arbitrary field $\mathbb{F}$, and $n \in \mathbb{N}$. A vector of the form

$$\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n,$$

where $\alpha_i \in \mathbb{F}$ and $v_i \in V$, for $i = 1, \ldots, n$, is called a *linear combination* of the vectors $v_1, v_2, \ldots, v_n$. The scalar α_i is called the *coefficient* of v_i, for $i = 1, \ldots, n$.

The set of all linear combinations of given vectors $v_1, v_2, \ldots, v_n \in V$ is a subspace of V and is called the *subspace generated by* (or *spanned by*) the vectors $v_1, v_2, \ldots, v_n$.

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a non-empty subset of V and $v \in V$. If there are scalars $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}$ such that

$$v = \alpha_1 s_1 + \alpha_2 s_2 + \ldots + \alpha_m s_n,$$

then one says that v *can be written as a linear combination* of the vectors in S. The set S is *linearly independent* if and only if no vector in S can be written as a linear combination of the other vectors in that set. If one vector in S can be written as a linear combination of the others, then the set of vectors is said to be *linearly dependent*.

A non-empty subset B of V is said to be a *basis* of V if and only if both of the following are true:

- B is a linearly independent set;

- V is spanned by B.

Example 2.4. *It is easy to see that the set* $\{(1,0,0);(0,1,0);(0,0,1)\}$ *is a basis of* $\mathbb{R}^3$*, which is called the* standard basis *of* $\mathbb{R}^3$*.*

A general concept of standard basis for vector subspaces will be given later in this chapter.

If V is a vector space that has a basis B containing a finite number of vectors, then V is said to be *finite dimensional*. The number of elements in that basis is what is called the *dimension* of V, and is denoted by $\dim(V)$. It can be shown that the dimension of a vector space does not depend on the basis chosen, since all the bases have the same number of elements [Val93]. If V has no finite basis, then V is said to be *infinite dimensional*.

Example 2.5. *From the previous example, it is clear that* $\mathbb{R}^3$ *is finite dimensional and* $\dim(\mathbb{R}^3) = 3$*.*

2.4 Matrices and Smith Normal Form

Let $m, n \in \mathbb{N}$ and R a commutative ring. Let $a_{i,j} \in R$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The rectangular array A defined by

$$A = [a_{i,j}] = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{2.1}$$

is called a *matrix over* R with m rows and n columns, or simply an $m \times n$ *matrix*. If $m = n$ one says that A is a *square matrix*. If $m \neq n$, then the matrix is said to be *non-square*. The set of all matrices over R with m rows and n columns is denoted by $\mathcal{M}_{m\times n}(R)$. If $m = n$, one denotes $\mathcal{M}_{n\times n}(R)$ simply by $\mathcal{M}_n(R)$. The elements of a

matrix are called its *entries*, and $a_{i,j}$ denotes the entry that occurs at the intersection of the ith row and jth column.

A matrix in $\mathcal{M}_{m\times n}(R)$ ($\mathcal{M}_n(R)$) in which each element is the additive identity of R is called a *zero matrix*, or *null matrix*, and is usually denoted by $\mathbf{0_{m\times n}}$ ($\mathbf{0_n}$).

Example 2.6. *The null matrices in $\mathcal{M}_3(R)$ and $\mathcal{M}_{2\times 4}(R)$ are, respectively,*

$$\mathbf{0}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \textit{and} \quad \mathbf{0}_{2\times 4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The $n \times n$ matrix $A = [a_{i,j}]$ over R such that $a_{i,i} = 1$ and $a_{i,j} = 0$, for $i \neq j$, is called the *identity matrix* of order n over R and is denoted by I_n.

Example 2.7. *The identity matrix of order 2 is* $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

An $m \times n$ matrix $A = [a_{i,j}]$ can be thought of either as a collection of m row vectors, each having n coordinates:

$$\begin{matrix} [a_{1,1} & a_{1,2} & \dots & a_{1,n}], \\ [a_{2,1} & a_{2,2} & \dots & a_{2,n}], \\ & & \vdots & \\ [a_{m,1} & a_{m,2} & \dots & a_{m,n}], \end{matrix}$$

or as a collection of n column vectors, each having m coordinates:

$$\begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{m,1} \end{bmatrix}, \begin{bmatrix} a_{1,2} \\ a_{2,2} \\ \vdots \\ a_{m,2} \end{bmatrix}, \dots, \begin{bmatrix} a_{1,n} \\ a_{2,n} \\ \vdots \\ a_{m,n} \end{bmatrix}.$$

The subspace of R^n generated by the row vectors of A is called the *row space* of the

matrix A. The dimension of this row space is called the *row rank* of A. Similarly, the subspace of R^m generated by the column vectors of A is called the *column space* of A, and its dimension is the *column rank* of A.

It is well known that the row rank of a matrix is equal to its column rank [McC71]. Therefore, one does not need to distinguish between the row rank and the column rank of a matrix. Accordingly, we make the following definition. The common value of the row rank and the column rank of a matrix is called simply the *rank* of the matrix. The rank of a matrix A is here denoted by $\text{rank}(A)$.

A matrix is said to have *maximal rank* if its rank equals the lesser of the number of rows and columns.

Example 2.8. *Consider the matrices*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \textit{and} \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

defined over $\mathbb{F}_2$. *Then, since* $\text{rank}(A) = 2 =$ *number of rows, we can say that* A *has maximal rank. The matrix* B *does not have maximal rank because* $\text{rank}(B) = 1 <$ *number of rows* $<$ *number of columns.*

One can define two operations that give $\mathcal{M}_n(R)$ a ring structure. Let $A = [a_{i,j}]$ and $B = [b_{i,j}]$ be matrices in $\mathcal{M}_{m\times n}(R)$. The sum of A and B is the $m \times n$ matrix $C = [c_{i,j}] = A + B$ such that

$$c_{i,j} = a_{i,j} + b_{i,j}.$$

Now, let $A = [a_{i,j}]$ be a matrix in $\mathcal{M}_{m\times n}(R)$ and $B = [b_{i,j}]$ a matrix in $\mathcal{M}_{n\times p}(R)$. The matrix product $C = [c_{i,j}] = AB$ is the $m \times p$ matrix defined by

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}.$$

The set $\mathcal{M}_n(R)$ together with the two operations defined above is a ring, which is not commutative. Notice that the addition of matrices is defined only for matrices of the same size, and the product is defined between matrices such that the number of columns of the first matrix equals the number of rows of the second one.

Example 2.9. *Consider the matrices A and B of the previous example. Then*

$$A + B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

and the product AB is not defined.

One can also define a scalar multiplication which, together with the matrix addition defined above, gives $\mathcal{M}_{m\times n}(R)$ a vector space structure. Let $\alpha \in R$ and let $A = [a_{i,j}]$ be an $m \times n$ matrix over R. Then, the matrix $C = [c_{i,j}] = \alpha A$, the scalar multiplication of α and A, is given by

$$c_{i,j} = \alpha a_{i,j}.$$

In this work we deal with several kinds of matrices. For example, we deal with matrices in $\mathcal{M}_{m\times n}(\mathbb{F})$ and with matrices in $\mathcal{M}_{m\times n}(\mathbb{F}[x])$, where $m, n \in \mathbb{N}$ and $\mathbb{F}$ is a finite field. Note that, unless $m = n$, those sets are not rings with the usual operations of addition and multiplication of matrices. The matrices in $\mathcal{M}_{m\times n}(\mathbb{F}[x])$ are called *polynomial matrices*, and there is a natural bijection between this set and the set of polynomials in x whose coefficients are $m \times n$ matrices over $\mathbb{F}$, *i.e.*, $\mathcal{M}_{m\times n}(\mathbb{F})[x]$. The elements of $\mathcal{M}_{m\times n}(\mathbb{F})[x]$ are called *matrix polynomials*.

Example 2.10. *Let $p(x)$ be the matrix polynomial in $\mathcal{M}_{2\times 3}(\mathbb{F}_2)[x]$ defined by*

$$p(x) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} x^2.$$

Then, the corresponding polynomial matrix in $\mathcal{M}_{2\times 3}(\mathbb{F}_2[x])$ *is*

$$P = \begin{bmatrix} 1+x^2 & 1 & 1 \\ x^2 & 0 & 1+x^2 \end{bmatrix}.$$

If A is an $m \times n$ matrix, then the *transpose* matrix of A is denoted by A^T and is the $n \times m$ matrix whose (i, j)th entry is the same as the (j, i)th entry of the original matrix A.

Example 2.11. *Let* A *and* B *be the following matrices over* $\mathbb{R}$*:*

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \textit{and} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}.$$

Then,

$$A^{\mathrm{T}} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \quad \textit{and} \quad B^{\mathrm{T}} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}.$$

For an $m \times n$ matrix A, the submatrix $A_{i,j}$ is obtained by deleting the ith row and the jth column of A.

Example 2.12. *Consider the matrix* B *of the previous example. Then* $B_{1,2} = [4, 6]$.

With each $n \times n$ matrix $A = [a_{i,j}]$ there is associated a unique number called the *determinant* of A and written $\det(A)$ or $|A|$. The determinant of A can be computed recursively as follows:

1. $|A| = a_{1,1}$, if $n = 1$;
2. $|A| = a_{1,1}a_{2,2} - a_{1,2}a_{2,1}$, if $n = 2$;
3. $|A| = \sum_{j=1}^{n} (-1)^{1+j} a_{1,j}|A_{1,j}|$, if $n > 2$.

It is well known that an $n \times n$ matrix A has rank n if and only if the determinant of A is not zero [McC71].

For an $n \times n$ matrix A, the *adjoint matrix* of A is the matrix

$$\mathrm{adj}(A) = [c_{i,j}],$$

where

$$c_{i,j} = (-1)^{i+j} \det(A_{j,i}).$$

Example 2.13. *Consider the matrices*

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \textit{and} \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

defined over $\mathbb{F}_2$. *Then,* $\det(A) = 1$, $\det(B) = 0$,

$$\mathrm{adj}(A) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad \textit{and} \quad \mathrm{adj}(B) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Let A to be an $n \times n$ matrix. A is called *invertible* (also *non-singular*) if there exists an $n \times n$ matrix B such that

$$AB = BA = I_n.$$

If this is the case, the matrix B is uniquely determined by A and is called the *inverse* of A, denoted by A^{-1}. The inverse of A can be computed in several ways. For example,

$$A^{-1} = \frac{1}{\det(A)} \mathrm{adj}(A).$$

Furthermore, A is invertible if and only if $\det(A) \neq 0$ or, equivalently, $\mathrm{rank}(A) = n$ [McC71]. The set of all $n \times n$ invertible matrices over R is denoted by $GL_n(R)$, which stands for general linear group of degree n over R.

Example 2.14. *The matrix B of the previous example is not invertible, while the matrix A is invertible and $A^{-1} = \mathrm{adj}(A)$.*

Proposition 2.15 ([MP13])**.** *Let $\mathbb{F}_q$ be a finite field with $q \in \mathbb{N}$ elements and $n \in \mathbb{N}$. Then*

$$|GL_n(\mathbb{F}_q)| = \prod_{i=0}^{n-1}(q^n - q^i).$$

Notice that non-square matrices are not invertible. However, they can be left or right invertible. An $m \times n$ matrix A is *left* (*right*) *invertible* if there is an $n \times m$ matrix B such that $BA = I_n$ ($AB = I_m$). Such a matrix B is called a *left* (*right*) *inverse* of A. One knows that A is left (right) invertible if and only if $\mathrm{rank}(A) = n$ ($\mathrm{rank}(A) = m$), *i.e.*, the columns (rows) of A are linearly independent. One says that a matrix is in *reduced row echelon form* if and only if all the following conditions hold:

- the first non-zero entry in each row is 1;
- each row has its first non-zero entry in a later column than any previous rows;
- all entries above and below the first non-zero entry of each row are zero;
- all rows having nothing but zeros are below all other rows of the matrix.

The matrix is said to be in *reduced column echelon form* if its transpose matrix is in reduced row echelon form.

Example 2.16. *The following matrix over $\mathbb{F}_2$ is in reduced row echelon form but is not in reduced column echelon form:*

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Let A and B be two matrices with the same number of rows. We define the augmented matrix $[A|B]$ as the matrix obtained by appending the columns of the matrices A and B.

Example 2.17. *If A and B are the following matrices over $\mathbb{R}$:*

$$A = \begin{bmatrix} 1 & 2 & 9 \\ -3 & 7 & 0 \end{bmatrix} \quad \textit{and} \quad B = \begin{bmatrix} 10 & 0 \\ 4 & 5 \end{bmatrix},$$

then

$$[A|B] = \begin{bmatrix} 1 & 2 & 9 & 10 & 0 \\ -3 & 7 & 0 & 4 & 5 \end{bmatrix}.$$

The following three results play an important role in Chapters 3 and 4.

Lemma 2.18. *Let $A \in \mathcal{M}_{m\times k}$, and $B \in \mathcal{M}_{m\times \ell}$. Then,* $\operatorname{rank}([A|B]) = \operatorname{rank}(A)$ *if and only if there is a matrix $X \in \mathcal{M}_{k\times\ell}$ such that $B = AX$.*

Proof. One knows that $\operatorname{rank}([A|B]) = \operatorname{rank}(A)$ if and only if the column space of A contains the column space of B. Since right multiplication by a matrix performs linear combinations on the columns of a matrix, it follows that there is a matrix $X \in \mathcal{M}_{k\times\ell}$ such that $B = AX$. □

Lemma 2.19. *Let $A, B \in \mathcal{M}_{m\times k}$. Then,* $\operatorname{rank}(A) = \operatorname{rank}([A|B]) = \operatorname{rank}(B)$ *if and only if there is a matrix $X \in GL_k$ such that $B = AX$.*

Proof. Let A' be the reduced column echelon form of A, and B' the reduced column echelon form of B. Let $X_A \in GL_k$ be the matrix such that $A' = AX_A$, and $X_B \in GL_k$ be the matrix such that $B' = BX_B$. Since $\operatorname{rank}(A) = \operatorname{rank}([A|B]) = \operatorname{rank}(B)$ if and only if $A' = B'$, one gets that $\operatorname{rank}(A) = \operatorname{rank}([A|B]) = \operatorname{rank}(B)$ if and only if $AX_A = BX_B$. That is, $AX_AX_B^{-1} = B$. Therefore, $\operatorname{rank}(A) = \operatorname{rank}([A|B]) = \operatorname{rank}(B)$ if and only if there is a matrix $X = X_AX_B^{-1} \in GL_k$ such that $B = AX$. □

Theorem 2.20. *Let $\mathbb{F}_q$ be a finite field with $q \in \mathbb{N}$ elements, $m, n \in \mathbb{N}$, and $A \in \mathcal{M}_{m\times n}(\mathbb{F}_q)$. Then, the number of distinct matrices of the form AX, where $X \in GL_n(\mathbb{F}_q)$ is*

$$\prod_{i=0}^{\operatorname{rank}(A)-1} (q^n - q^i).$$

Proof. Let $A \in \mathcal{M}_{m\times n}(\mathbb{F}_q)$. We show that the number of matrices $X \in GL_n(\mathbb{F}_q)$ such that $AX = A$ is $\prod_{i=\mathrm{rank}(A)}^{n-1}(q^n - q^i)$, when $\mathrm{rank}(A) \neq n$, and equals 1 when $\mathrm{rank}(A) = n$. The result then follows from the well-known size of $GL_n(\mathbb{F}_q)$ (given in Proposition 2.15).

Let $X \in GL_n(\mathbb{F}_q)$ be such that $AX = A$. Then, there are $n - \mathrm{rank}(A)$ rows in X whose entries can be arbitrarily chosen to have a solution of $AX = A$. But, since X has to be invertible, one has $q^n - q^{\mathrm{rank}(A)}$ possibilities for the first of those rows, $q^n - q^{\mathrm{rank}(A)+1}$ for the second, $q^n - q^{\mathrm{rank}(A)+2}$ for the third, and so on. Therefore, there are $(q^n - q^{\mathrm{rank}(A)})(q^n - q^{\mathrm{rank}(A)+1})\cdots(q^n - q^{n-1})$ matrices X that satisfy the required condition. □

Let V be a vector subspace of $\mathbb{F}^n$ with dimension k, where $\mathbb{F}$ is a field and $n \in \mathbb{N}$. The unique basis $\{b_1, b_2, \ldots, b_k\}$ of V such that the matrix $[b_1\ b_2\ \cdots\ b_k]$ is in reduced column echelon form will be here referred to as the *standard basis* of V.

Two $m \times n$ matrices A, B, with entries in a PID, R, are said to be *equivalent* if there exist matrices $P \in GL_m(R)$ and $N \in GL_n(R)$ such that $B = PAN$.

It is clear that matrix equivalence is an equivalence relation in the set $\mathcal{M}_{m\times n}(R)$.

The following result is well known (see [Jac85] or [New72, Theorem II.9]).

Theorem 2.21. *Let R be a principal ideal domain. Every matrix $A \in \mathcal{M}_{m\times n}(R)$ is equivalent to a matrix of the form*

$$\mathcal{D} = diag(d_1, d_2, \ldots, d_r, 0, \ldots, 0) = \begin{bmatrix} d_1 & & & & & \\ & \ddots & & & 0 & \\ & & d_r & & & \\ & & & 0 & & \\ & 0 & & & \ddots & \\ & & & & & 0 \end{bmatrix}$$

where r is the rank of A, $d_i \neq 0$ and $d_i \mid d_{i+1}$, i.e. d_i divides d_{i+1}, for $1 \leq i \leq r-1$. The matrix $\mathcal{D}$ is called the Smith normal form *of A, denoted* $\mathrm{SNF}(A)$*, and the elements d_i*

are called the invariant factors *of* A.

Example 2.22. *The Smith normal form of the matrix*

$$A = \begin{bmatrix} x^2 & 1 \\ 1+x & 0 \\ 0 & x+x^2 \end{bmatrix},$$

defined over $\mathbb{F}_2[x]$*, is*

$$\mathrm{SNF}(A) = (1, 1+x) = \begin{bmatrix} 1 & 0 \\ 0 & 1+x \\ 0 & 0 \end{bmatrix},$$

and the matrices $P \in GL_3(\mathbb{F}_2[z])$ *and* $N \in GL_2(\mathbb{F}_2[z])$ *such that* $\mathrm{SNF}(A) = PAN$ *are*

$$P = \begin{bmatrix} 1 & 1+x & 0 \\ 1+x & x^2 & 0 \\ x+x^2 & x^3 & 1 \end{bmatrix} \quad \textit{and} \quad N = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

2.5 Cayley-Hamilton Theorem and Some Implications

Let $n \in \mathbb{N}$, A an $n \times n$ matrix over a field $\mathbb{F}$, and I_n the $n \times n$ identity matrix over the same field. The *characteristic polynomial* of A is defined as

$$p_A(\lambda) = \det(\lambda I_n - A).$$

Since the entries of the matrix $\lambda I_n - A$ are linear or constant polynomials in λ, its determinant is a monic polynomial in λ of order n. Therefore, the degree of the characteristic polynomial of a $n \times n$ matrix is n.

Example 2.23. *The characteristic polynomial of the square matrix*

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

defined over $\mathbb{F}_2$*, is*

$$p_A(\lambda) = \begin{vmatrix} \lambda+1 & 1 & 0 \\ 1 & \lambda & 1 \\ 0 & 1 & \lambda \end{vmatrix} = 1 + \lambda^2 + \lambda^3.$$

Theorem 2.24 (Cayley-Hamilton [Val93])**.** *If A is an $n \times n$ matrix over an arbitrary field $\mathbb{F}$, and p is the characteristic polynomial of A, then*

$$p(A) = 0.$$

The Cayley-Hamilton theorem allows us to express A^n as a linear combination of the lower powers of A.

Example 2.25. *Considering the matrix A from the previous example, one has*

$$1 + A^2 + A^3 = \mathbf{0}_3 \iff A^3 = 1 + A^2.$$

The *minimal polynomial* of an $n \times n$ matrix A over a field $\mathbb{F}$ is the monic polynomial m over $\mathbb{F}$ of least degree such that $m(A) = 0$.

Any other polynomial q with $q(A) = 0$ is a multiple of m. Therefore, since the characteristic polynomial p of A has degree n and $p(A) = 0$, it follows that the minimal polynomial of A has degree at most n. These observations will be fundamental in Chapter 3.

2.6 Linear Maps

Let V and W be vector spaces over the same field $\mathbb{F}$. A mapping $f : V \to W$ is called a *linear transformation*, *linear map* or an *homomorphism* of V into W, if the following conditions are true:

- $f(v_1 + v_2) = f(v_1) + f(v_2)$, for all v_1, v_2 in V;
- $f(\alpha v) = \alpha f(v)$, for all α in $\mathbb{F}$ and for all v in V.

The first condition states that addition is preserved under the mapping f. The second asserts that also scalar multiplication is preserved under the mapping f. This is equivalent to require that the same happens for any linear combination of vectors, *i.e.*, that for any vectors $v_1, \ldots, v_n \in V$, and scalars $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$, the following equality holds:

$$f(\alpha_1 v_1 + \cdots + \alpha_n v_n) = \alpha_1 f(v_1) + \cdots + \alpha_n f(v_n).$$

Denoting the zero elements of the vector spaces V and W by 0_V and 0_W respectively, it follows that $f(0_V) = 0_W$ because letting $\alpha = 0$ in the second condition one gets:

$$f(0_V) = f(0 \cdot 0_V) = 0 f(0_V) = 0_W.$$

An homomorphism which is a bijective mapping is called a *linear isomorphism*, and if there exists an isomorphism φ of V onto W we say that V is *isomorphic* to W, denoted by $V \simeq W$, and φ is called a *vector space isomorphism*.

If V and W are finite dimensional vector spaces, and an ordered basis is defined for each vector space, then every linear map from V to W can be represented by a matrix. Moreover, matrices yield examples of linear maps. For example, if A is an $m \times n$ matrix over a ring R, then A defines a linear map from R^n to R^m by sending the column vector $v \in R^n$ to the column vector $Av \in R^m$.

Now, let us see how to construct the matrix of a linear map. Let $m, n \in \mathbb{N}$ be the dimensions of the vector spaces V and W, respectively. Let $f : V \to W$ be a linear transformation and let $\mathcal{B}_V = \{v_1, \ldots, v_m\}$ be a basis for V. Then, every vector v in V is uniquely determined by the coefficients $\alpha_1, \ldots, \alpha_m$ in $\mathbb{F}$ such that

$$v = \alpha_1 v_1 + \cdots + \alpha_m v_m.$$

Since f is a linear map, one has:

$$f(\alpha_1 v_1 + \cdots + \alpha_m v_m) = \alpha_1 f(v_1) + \cdots + \alpha_m f(v_m),$$

which implies that the function f is entirely determined by the vectors $f(v_1), \ldots, f(v_m)$. Now let $\mathcal{B}_W = \{w_1, \ldots, w_n\}$ be a basis for W. Then, we can represent each vector $f(v_j)$, for $j = 1, \ldots, m$, as

$$f(v_j) = a_{1,j} w_1 + \cdots + a_{m,j} w_m.$$

Thus the function f is entirely determined by the values of $a_{i,j}$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. If we put these values into an $m \times n$ matrix M, then we can conveniently use it to compute the vector output of f for any vector v in V. To obtain M, every column j of M is a vector

$$\begin{bmatrix} a_{1,j} \\ \vdots \\ a_{m,j} \end{bmatrix}$$

corresponding to $f(v_j)$ as defined above. In other words, every column $j = 1, \ldots, n$ has a corresponding vector $f(v_j)$ whose coordinates $a_{1j}, \ldots, a_{m,j}$ are the elements of that column. The matrix constructed in this way is called the *matrix of the linear application relative to the bases* $\mathcal{B}_V$ and $\mathcal{B}_W$. Left multiplication by A takes a vector written in terms of $\mathcal{B}_V$, applies f, and writes the result in terms of $\mathcal{B}_W$. It is then obvious that a linear map may be defined by many matrices, since the values of the elements of a matrix depend on the bases chosen.

Below we present an example where we compute the matrix of a linear application relative to the standard bases of the vector spaces considered. This is the simplest case, but is also the most relevant for this work.

Example 2.26. *Let $f : \mathbb{F}_2^3 \to \mathbb{F}_2^2$ be the mapping defined by:*

$$f(x, y, z) = (x + y, z).$$

First, let us see that f is linear.

1. *Let $v = (v_1, v_2, v_3), w = (w_1, w_2, w_3) \in \mathbb{F}_2^3$. Then*

$$\begin{aligned} f(v + w) &= f(v_1 + w_1, v_2 + w_2, v_3 + w_3) \\ &= (v_1 + w_1 + v_2 + w_2, v_3 + w_3) \\ &= (v_1 + v_2, v_3) + (w_1 + w_2, w_3) \\ &= f(v) + f(w). \end{aligned}$$

2. *Let $\alpha \in \mathbb{F}_2$ and $v = (v_1, v_2, v_3) \in \mathbb{F}_2^3$. Then*

$$\begin{aligned} f(\alpha v) &= f(\alpha v_1, \alpha v_2, \alpha v_3) \\ &= (\alpha v_1 + \alpha v_2, \alpha v_3) \\ &= \alpha(v_1 + v_2, v_3) \\ &= \alpha f(v). \end{aligned}$$

Since addition and scalar multiplication are preserved under f, one concludes that f is a linear map.

Now, let $\mathcal{B}$ be the standard basis of $\mathbb{F}_2^3$, i.e.,

$$\mathcal{B} = \{(1, 0, 0); (0, 1, 0); (0, 0, 1)\}.$$

One has,

$$f(1,0,0) = (1,0)$$
$$f(0,1,0) = (1,0)$$
$$f(0,0,1) = (0,1).$$

Therefore, the matrix of f relative to $\mathcal{B}$ and the standard basis of $\mathbb{F}_2^2$ is

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and, for example,

$$f(1,1,0) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Given a matrix, A, of a linear application, f, it is well known that if the rows (columns) of A are linearly independent, then f is surjective (injective).

Example 2.27. *The mapping f defined in the previous example is surjective, because the matrix of the application has linearly independent rows.*

Chapter 3

Linear Finite Transducers

3.1 Preliminaries on Finite Transducers

In what follows, an *alphabet* is a non-empty finite set of elements. The elements of an alphabet are called *symbols* or *letters*. Given an alphabet A, a finite sequence of symbols from A, say $a_0a_1\cdots a_{\ell-1}$, is called a *word* over A, and ℓ its *length*. When $\ell = 0$, the sequence $a_0a_1\cdots a_{\ell-1}$ is an empty sequence which contains no element and it is called the *empty word*. We use ε to denote the empty word, and $|\alpha|$ to denote the length of the word α. We let A^n be the set of words of length n, where $n \in \mathbb{N}_0$, and $A^0 = \{\varepsilon\}$. We put $A^\star = \cup_{n\geq 0}A^n$, the set of all finite words, and $A^\omega = \{a_0a_1\cdots a_n\cdots \mid a_i \in A\}$ is the set of infinite words.

Let $\alpha = a_0a_1\cdots a_{m-1}$ and $\beta = b_0b_1\cdots b_{n-1}$ be two words in $A^\star$ of length m and n, respectively. The concatenation of α and β is $a_0a_1\cdots a_{m-1}b_0b_1\cdots b_{n-1}$, which is also a word in $A^\star$, of length $m+n$, and is denoted by $\alpha\beta$. Clearly, $\alpha\varepsilon = \varepsilon\alpha = \alpha$. Similarly, if $\alpha = a_0a_1\cdots a_{m-1} \in A^\star$ and $\beta = b_0b_1\cdots b_{n-1}\cdots \in A^\omega$, then the concatenation of α and β is the element $a_0a_1\cdots a_{m-1}b_0b_1\cdots b_{n-1}\cdots$ of A^ω. It is obvious that $\varepsilon\beta = \beta$. For any $U, V \subseteq A^\star$, the concatenation of U and V is the set $\{\alpha\beta \mid \alpha \in U, \beta \in V\}$.

In the context of this work, a *finite transducer* (FT) is a deterministic finite state

sequential machine which, in any given state, reads a symbol from a set $\mathcal{X}$, produces a symbol from a set $\mathcal{Y}$, and switches to another state. Thus, given an initial state and a finite input sequence, a transducer produces an output sequence of the same length. The formal definition of a finite transducer is the following.

Definition 3.1. *A* finite transducer *is a quintuple* $\langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$*, where:*

- $\mathcal{X}$ *is a non-empty finite set, called the* input alphabet*;*
- $\mathcal{Y}$ *is a non-empty finite set, called the* output alphabet*;*
- S *is a non-empty finite set called the* set of states*;*
- $\delta : S \times \mathcal{X} \to S$*, called the* state transition function*;*
- $\lambda : S \times \mathcal{X} \to \mathcal{Y}$*, called the* output function*.*

These transducers are deterministic and can be seen as having all the states as final. Every state in S can be used as initial, and this gives rise to a determinist transducer in the usual sense, also known as Mealy machine [Sta72, Rut06]. Therefore, in what follows, a transducer is a family of classical transducers that share the same underlying digraph.

Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer. The state transition function δ and the output function λ can be extended to finite words, *i.e.*, elements of $\mathcal{X}^\star$, recursively, as follows:

$$\begin{aligned} \delta(s, \varepsilon) &= s, & \delta(s, x\alpha) &= \delta(\delta(s, x), \alpha), \\ \lambda(s, \varepsilon) &= \varepsilon, & \lambda(s, x\alpha) &= \lambda(s, x)\, \lambda(\delta(s, x), \alpha), \end{aligned}$$

where $s \in S$, $x \in \mathcal{X}$, and $\alpha \in \mathcal{X}^\star$. In an analogous way, λ may be extended to $\mathcal{X}^\omega$.

From these definitions it follows that one has, for all $s \in S, \alpha, \beta \in \mathcal{X}^\star$,

$$\delta(s, \alpha\beta) = \delta(\delta(s, \alpha), \beta)$$

and, for all $s \in S, \alpha \in \mathcal{X}^\star$, $\beta \in \mathcal{X}^\star \cup \mathcal{X}^\omega$,

$$\lambda(s, \alpha\beta) = \lambda(s, \alpha)\, \lambda(\delta(s, \alpha), \beta).$$

Example 3.2. *Let $M = \langle \{0,1\}, \{a,b\}, \{s_1, s_2\}, \delta, \lambda \rangle$ be the transducer defined by:*

$$\begin{array}{llll} \delta(s_1,0) = s_1, & \delta(s_1,1) = s_2, & \delta(s_2,0) = s_1, & \delta(s_2,1) = s_2, \\ \lambda(s_1,0) = a, & \lambda(s_1,1) = a, & \lambda(s_2,0) = b, & \lambda(s_2,1) = b. \end{array}$$

Then, for example,

$$\begin{aligned} \delta(s_1, 01) &= \delta(\delta(s_1,0),1) = \delta(s_1,1) = s_2, \\ \lambda(s_1, 01) &= \lambda(s_1,0)\lambda(\delta(s_1,0),1) = a\lambda(s_1,1) = aa, \end{aligned}$$

and

$$\begin{aligned} \delta(s_1, 0010110) &= s_1, \\ \lambda(s_1, 0010110) &= aaababb. \end{aligned}$$

Example 3.3. *Let $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, \mathbb{F}_2^2, \delta, \lambda \rangle$ be the transducer defined by:*

$$\begin{aligned} \delta(s, x) &= As + Bx, \\ \lambda(s, x) &= Cs + Dx, \end{aligned}$$

for all $s \in \mathbb{F}_2^2$, $x \in \mathbb{F}_2^2$, and where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},\ C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Take $s = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ *and* $\alpha = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. *Then,*

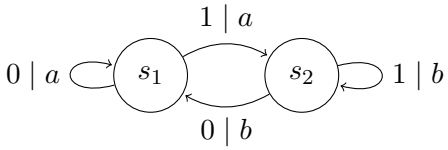
$$\delta(s, \alpha) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\lambda(s, \alpha) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

M is what is called a linear finite transducer. The formal definition will be given in Section 3.2.

A transducer can be represented by a diagram that is a digraph with labeled nodes and arcs, where loops and multiple arcs are allowed. Each state of the transducer is represented by a node, and each arc indicates a transition between states. The label of each arc is a compound symbol of the form $i \mid o$, where i and o stand for the input and output symbol, respectively. This representation is useful to deal by hand with the computations of some examples presented in this chapter.

Example 3.4. *The transducer M defined in Example 3.2 is represented by the diagram below.*



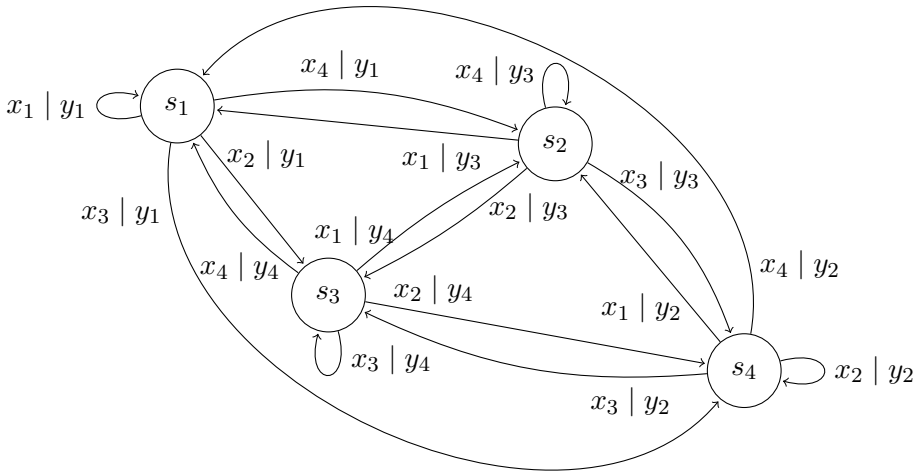
Example 3.5. *Let*

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$s_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad s_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad s_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad s_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$y_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad y_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad y_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad y_4 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Then, the transducer M of Example 3.3 is represented by the following diagram:



Given this diagram, it is quite easy to compute $\delta(s, \alpha)$ and $\lambda(s, \alpha)$, for the transducer defined in Example 3.3.

Definition 3.6. *Let $M_1 = \langle \mathcal{X}, \mathcal{Y}, S_1, \delta_1, \lambda_1 \rangle$ and $M_2 = \langle \mathcal{X}, \mathcal{Y}, S_2, \delta_2, \lambda_2 \rangle$ be two finite transducers. The transducers M_1 and M_2 are said to be* isomorphic*, and denoted $M_1 \simeq M_2$, if there exists a bijective map $\psi : S_1 \to S_2$ such that*

$$\psi(\delta_1(s_1, x)) = \delta_2(\psi(s_1), x), \text{ and}$$
$$\lambda_1(s_1, x) = \lambda_2(\psi(s_1), x),$$

for all $s_1 \in S_1$, and for all $x \in \mathcal{X}$. The map ψ is called an isomorphism *between M_1 and M_2.*

Remark 3.7. *Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer, S' a non-empty set, and $\psi : S \to S'$ a bijective map. The transducer $M_\psi = \langle \mathcal{X}, \mathcal{Y}, S', \delta_\psi, \lambda_\psi \rangle$ defined by*

$$\begin{aligned} \delta_\psi(s', x) &= \psi\left(\delta\left(\psi^{-1}(s'), x\right)\right), \\ \lambda_\psi(s', x) &= \lambda\left(\psi^{-1}(s'), x\right), \end{aligned}$$

for all $s' \in S'$, $x \in \mathcal{X}$, is isomorphic to M because ψ satisfies the two conditions in the previous definition:

$$\begin{aligned} \delta_\psi(\psi(s), x) &= \psi\left(\delta(\psi^{-1}(\psi(s)), x)\right) = \psi(\delta(s, x)); \\ \lambda_\psi(\psi(s), x) &= \lambda(\psi^{-1}(\psi(s)), x) = \lambda(s, x). \end{aligned}$$

Definition 3.8. *Let $M_1 = \langle \mathcal{X}, \mathcal{Y}_1, S_1, \delta_1, \lambda_1 \rangle$ and $M_2 = \langle \mathcal{X}, \mathcal{Y}_2, S_2, \delta_2, \lambda_2 \rangle$ be two finite transducers. Let $s_1 \in S_1$, and $s_2 \in S_2$. One says that s_1 and s_2 are* equivalent*, and denote this relation by $s_1 \sim s_2$, if*

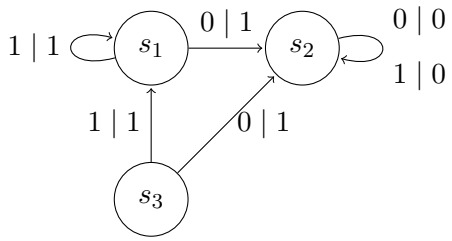
$$\forall \alpha \in \mathcal{X}^\star,\ \lambda_1(s_1, \alpha) = \lambda_2(s_2, \alpha).$$

It is obvious that if $s_1 \sim s_2$, then

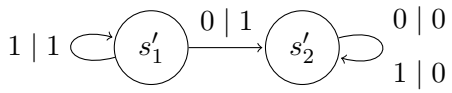
$$\forall x \in \mathcal{X},\ \delta_1(s_1, x) \sim \delta_2(s_2, x).$$

Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer. Trivially, the relation $\sim$ is an equivalence relation on S. As usual, we will denote by $[s]_\sim$ or $[s]$ the equivalence class that contains s, and by $S/\!\sim$ the set of equivalence classes of S, *i.e.*, $S/\!\sim = \{[s]_\sim \mid s \in S\}$.

Example 3.9. *Let $M = \langle \mathbb{F}_2, \mathbb{F}_2, \{s_1, s_2, s_3\}, \delta, \lambda \rangle$ be the transducer induced by the diagram:*



and let $M' = \langle \mathbb{F}_2, \mathbb{F}_2, \{s'_1, s'_2\}, \delta', \lambda' \rangle$ *be the transducer induced by:*



Then

- $s_2 \sim s'_2$, *because* $\forall \alpha \in \mathcal{X}^\star,\ \lambda(s_2, \alpha) = 0 \cdots 0 = \lambda'(s'_2, \alpha)$;
- $s_1 \sim s_3 \sim s'_1$.

To prove that $s_1 \sim s_3$, *let* α *be a non-empty word in* $\mathbb{F}_2^\star$. *Then, either* α *is of the form* 0β *or* α *is of the form* 1β, *for some* β *in* $\mathbb{F}_2^\star$. *In the first case, one has*

$$\lambda(s_1, 0\beta) = \lambda(s_1, 0)\lambda(\delta(s_1, 0), \beta) = 1\,\lambda(s_2, \beta),$$

and

$$\lambda(s_3, 0\beta) = \lambda(s_3, 0)\lambda(\delta(s_3, 0), \beta) = 1\,\lambda(s_2, \beta).$$

It follows that $\lambda(s_1, 0\beta) = \lambda(s_3, 0\beta)$, *for all* $\beta \in \mathcal{X}^\star$. *Analogously,*

$$\lambda(s_1, 1\beta) = 1\,\lambda(s_1, \beta) = \lambda(s_3, 1\beta),$$

for all $\beta \in \mathcal{X}^\star$. *Therefore,* $\forall \alpha \in \mathcal{X}^\star,\ \lambda(s_1, \alpha) = \lambda(s_3, \alpha)$, *i.e.,* $s_1 \sim s_3$. *It is also easy to see that* $s_1 \sim s'_1$.

Example 3.10. *Let* $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta, \lambda \rangle$ *be the transducer defined by:*

$$\delta(s, x) = As + Bx,$$
$$\lambda(s, x) = Cs + Dx,$$

for all $s \in \mathbb{F}_2^2$, $x \in \mathbb{F}_2^2$, *and where*

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ C = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

The pair of states $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ *and* $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ *are equivalent, as well as* $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ *and* $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

In Section 3.3 we present a method to directly check the equivalence of states for linear finite transducers. We now introduce the notion of equivalent transducers used in this context.

Definition 3.11. *Let* $M_1 = \langle \mathcal{X}, \mathcal{Y}_1, S_1, \delta_1, \lambda_1 \rangle$ *and* $M_2 = \langle \mathcal{X}, \mathcal{Y}_2, S_2, \delta_2, \lambda_2 \rangle$ *be two FTs.* M_1 *and* M_2 *are said to be* equivalent*, and denote this by* $M_1 \sim M_2$*, if the following two conditions are simultaneously satisfied:*

- $\forall s_1 \in S_1,\ \exists s_2 \in S_2 :\ s_1 \sim s_2$*;*
- $\forall s_2 \in S_2,\ \exists s_1 \in S_1 :\ s_1 \sim s_2$.

The relation $\sim$ defines an equivalence relation on the set of finite transducers.

Example 3.12. *The transducers* M *and* M' *of Example 3.9 are equivalent, since* $s_1 \sim s_3 \sim s_1'$ *and* $s_2 \sim s_2'$.

Definition 3.13. *Let* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *be a finite transducer. Considering the equivalence relation* $\sim$ *on the set of states, one defines the* quotient transducer $M/\!\sim = \langle \mathcal{X}, \mathcal{Y}, S/\!\sim, \delta_\sim, \lambda_\sim \rangle$ *in the following way*

$$\delta_\sim([s], x) = [\delta(s, x)] \quad \text{and} \quad \lambda_\sim([s], x) = \lambda(s, x),$$

for all $[s] \in S/\sim$, $x \in \mathcal{X}$.

Lemma 3.14. *$\delta_\sim$ and $\lambda_\sim$ are well defined.*

Proof. Let $[s_1], [s_2] \in S/\sim$ such that $[s_1] = [s_2]$. Since $[s_1] = [s_2]$ if and only if $s_1 \sim s_2$, it follows that

$$\delta(s_1, x) \sim \delta(s_2, x), \forall x \in \mathcal{X}, \text{ and } \lambda(s_1, \alpha) = \lambda(s_2, \alpha), \forall \alpha \in \mathcal{X}^\star.$$

Consequently,

$$[\delta(s_1, x)] = [\delta(s_2, x)], \forall x \in \mathcal{X}, \text{ and } \lambda(s_1, \alpha) = \lambda(s_2, \alpha), \forall \alpha \in \mathcal{X}^\star.$$

Thus, $\delta_\sim$ and $\lambda_\sim$ are well defined. □

Lemma 3.15. *For all $s \in S$, $\alpha \in \mathcal{X}^\star$, one has*

$$\delta_\sim([s], \alpha) = [\delta(s, \alpha)].$$

Proof. (by induction on the length of α)

The case $|\alpha| = 1$ is immediate from definition of $\delta_\sim$. Assume that, given $n \in \mathbb{N}$, the equality holds when $|\alpha| = n$. Let $x \in \mathcal{X}$. Then

$$\begin{aligned}
\delta_\sim([s], \alpha x) &= \delta_\sim(\delta_\sim([s], \alpha), x), \\
&= \delta_\sim([\delta(s, \alpha)], x), \text{ from hypothesis,} \\
&= [\delta(\delta(s, \alpha), x)], \text{ from the definition of } \delta_\sim, \\
&= [\delta(s, \alpha x)].
\end{aligned}$$

□

Lemma 3.16. *For all $s \in S$, one has $s \sim [s]$,* i.e.*,*

$$\forall \alpha \in \mathcal{X}^\star, \ \lambda(s, \alpha) = \lambda_\sim([s], \alpha).$$

Proof. (by induction on the length of α)

The case $|\alpha| = 1$ is obvious from definition of $\lambda_\sim$. Assume that, given $n \in \mathbb{N}$, the equality holds when $|\alpha| = n$. Let $x \in \mathcal{X}$. Then

$$\begin{aligned}
\lambda_\sim([s], \alpha x) &= \lambda_\sim(\lambda_\sim([s], \alpha), x), \\
&= \lambda(s, \alpha)\lambda_\sim([\delta(s, \alpha)], x), \text{ from hypothesis,} \\
&= \lambda(s, \alpha)\lambda(\delta(s, \alpha), x), \text{ from the definition of } \lambda_\sim, \\
&= \lambda(s, \alpha x).
\end{aligned}$$

□

Theorem 3.17. *Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer. Then, the quotient transducer $M/\sim = \langle \mathcal{X}, \mathcal{Y}, S/\sim, \delta_\sim, \lambda_\sim \rangle$ is equivalent to M.*

Proof. To prove that $M/\sim$ is equivalent to M, by definition, one needs to prove that:

1. $\forall s \in S,\ \exists s' \in S/\sim :\ s \sim s'$;
2. $\forall s' \in S/\sim,\ \exists s \in S :\ s \sim s'$.

To prove the first condition, one just needs to take $s' = [s]$, because, by Lemma 3.16, $s \sim [s]$. To prove the second condition, let $s' \in S/\sim$. Take $s \in S$ such that $s' = [s]$. Since, $s \sim [s]$, the condition follows. □

Definition 3.18. *A finite transducer is called* minimal *if it has no equivalent transducer with fewer states.*

Proposition 3.19. *A finite transducer is* minimal *if and only if it has no pair of equivalent states.*

Proof. Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer. We prove the "if part" by proving that if M is not minimal, then M has at least a pair of equivalent states. Assume that

M is not minimal. Then, by definition, there is a transducer $M' = \langle \mathcal{X}, \mathcal{Y}', S', \delta', \lambda' \rangle$ such that $M \sim M'$ and $|S'| < |S|$. From $M \sim M'$, it follows that

$$\forall s \in S,\ \exists s' \in S',\ s \sim s'.$$

Since $|S'| < |S|$, this implies that there are at least two states $s_1, s_2 \in S$ such that $s_1 \sim s' \sim s_2$, for some $s' \in S'$. Thus, M has at least a pair of equivalent states.

To prove the "only if" part, we prove that if M has at least a pair of equivalent states, then M is not minimal. Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer which has at least a pair of equivalent states. Then $|S/\!\sim| \leq |S| - 1$. Consequently, M is not minimal because $M/\!\sim$ is an equivalent transducer (by Theorem 3.17) with fewer states. □

Example 3.20. *The transducer M defined in Example 3.10 is equivalent to the transducer $M' = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, \mathbb{F}_2, \delta', \lambda' \rangle$ defined by:*

$$\begin{aligned} \delta'(s', x) &= A's' + B'x, \\ \lambda'(s', x) &= C's' + D'x, \end{aligned}$$

for all $s' \in \mathbb{F}_2^2$, $x \in \mathbb{F}_2^2$, and where

$$A' = \begin{bmatrix} 0 \end{bmatrix},\ B' = \begin{bmatrix} 1 & 0 \end{bmatrix},\ C' = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \text{and } D' = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

First notice that M' has only two states, $s'_1 = 0$ and $s'_2 = 1$, which are not equivalent since, for example,

$$\lambda' \left(s'_1, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \lambda' \left(s'_2, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right).$$

Therefore M' is minimal. It can also be shown that $s'_1 \sim s_1 \sim s_2$ and $s'_1 \sim s_3 \sim s_4$, where $s_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $s_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $s_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and $s_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

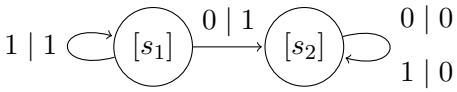
Example 3.21. *The transducer M of Example 3.9 is not minimal because states s_1 and s_3 are equivalent. The transducer M', in the same example, is minimal because s'_1 and s'_2 are not equivalent (for example, $\lambda'(s'_1, 0) = 1 \neq 0 = \lambda'(s'_2, 0)$).*

Theorem 3.22. *Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a finite transducer. The transducer $M/\sim$ is minimal.*

Proof. Since $M/\sim$ is constructed in a way that all states equivalent to a given state in M are collapsed into a single state of $S/\sim$, one concludes that $M/\sim$ has no pair of equivalent states, *i.e.*, $M/\sim$ is minimal. □

From the previous theorem, constructing the quotient transducer of a finite transducer M is a method to obtain a minimal FT equivalent to M. In Section 3.4, we adapt this method to minimize linear finite transducers.

Example 3.23. *Consider the transducer M of Example 3.9. One knows that $s_1 \sim s_3$. Then, $M/\sim$ is the transducer induced by the diagram below.*



The transducer $M/\sim$ is minimal and isomorphic to the transducer M' also presented in Example 3.9 (it is quite obvious that the application $\psi : \{[s_1], [s_2]\} \to \{s'_1, s'_2\}$ defined by $\psi([s_1]) = s'_1$ and $\psi([s_2]) = s'_2$ is an isomorphism between $M/\sim$ and M').

It is clear that if $M_1 \simeq M_2$, then $M_1 \sim M_2$. Conversely, if M_1 and M_2 are minimal and equivalent, and $\mathcal{Y}_1 = \mathcal{Y}_2$, then it can be proven that M_1 and M_2 are isomorphic. Just consider ψ to be the relation $\sim$ from S_1 to S_2 [Tao09, page 11]. Thus, a minimal transducer is unique up to isomorphism.

3.1.1 Concepts on Invertibility

A fundamental concept in this work is the concept of injectivity that is behind the invertibility property of the transducers used for cryptographic purposes. In fact, we will talk about two concepts: the concept of ω-injectivity and the concept of injectivity with a certain delay. These two notions of injectivity were introduced, as far as we know, by Tao, who called them *weakly invertible* and *weakly invertible with a certain delay*, respectively [Tao09]. Here we use names that are more naturally related to how these terms are used in other mathematical settings.

Definition 3.24. *A finite transducer* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *is* ω-injective*, if*

$$\forall s \in S,\ \forall \alpha, \alpha' \in \mathcal{X}^\omega, \quad \lambda(s, \alpha) = \lambda(s, \alpha') \ \Rightarrow\ \alpha = \alpha'.$$

That is, for any $s \in S$*, and any* $\alpha \in \mathcal{X}^\omega$*,* α *can be uniquely determined by* s *and* $\lambda(s, \alpha)$*.*

Definition 3.25. *A finite transducer* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *is* injective with delay τ*, or* τ-injective*, with* $\tau \in \mathbb{N}_0$*, if*

$$\forall s \in S,\ \forall x, x' \in \mathcal{X},\ \forall \alpha, \alpha' \in \mathcal{X}^\tau, \quad \lambda(s, x\alpha) = \lambda(s, x'\alpha') \ \Rightarrow\ x = x'.$$

That is, for any $s \in S$*,* $x \in \mathcal{X}$*, and* $\alpha \in \mathcal{X}^\tau$*,* x *is uniquely determined by* s *and* $\lambda(s, x\alpha)$*.*

To simplify, an equivalence class formed by ω-injective FTs is said to be ω-injective. Analogously, an equivalence class of τ-injective FTs, for some $\tau \in \mathbb{N}_0$, is said to be τ-injective.

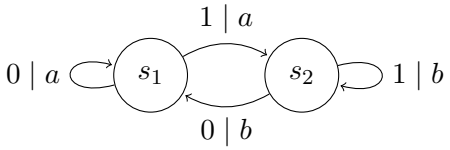
Later in this work, we deal with the case $\mathcal{X} = \mathbb{F}^\ell$, where $\mathbb{F}$ is a field, and it will be useful to identify the elements of $\mathcal{X}^\omega$ with the elements of $\mathbb{F}[[z]]^\ell$, by replacing $x_0 x_1 x_2 \cdots$ with $\sum_{i \geq 0} x_i z^i$. In that context, and from the definition of congruence modulo n relation,

a finite transducer $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ is injective with delay τ if and only if

$$\lambda(s, X) \equiv \lambda(s, X') \pmod{z^{\tau+1}} \;\Rightarrow\; X \equiv X' \pmod{z}, \tag{3.1}$$

for all $s \in S$, and $X, X' \in \mathbb{F}[[z]]^{\ell}$.

Example 3.26. *The transducer presented in Example 3.2, and which is represented by the diagram*



is injective with delay 1. To prove that, one has to compute the output for every state and every input sequence of length 2:

$$\lambda(s_1, 00) = aa, \qquad \lambda(s_2, 00) = ba, \qquad \lambda(s_1, 10) = ab, \qquad \lambda(s_2, 10) = bb,$$
$$\lambda(s_1, 01) = aa, \qquad \lambda(s_2, 01) = ba, \qquad \lambda(s_1, 11) = ab, \qquad \lambda(s_2, 11) = bb.$$

From these outputs, one can conclude that

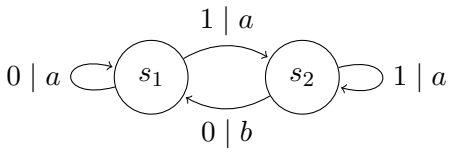
$$\forall\, s \in \{s_1, s_2\},\ \forall\, x_0\, x_1, x_0'\, x_1' \in \{0,1\}^2, \quad \lambda(s, x_0 x_1) = \lambda(s, x_0' x_1') \;\Rightarrow\; x_0 = x_0',$$

which proves, by definition, that the transducer is injective with delay 1. Moreover, the transducer is not injective with delay 0 (for example, $\lambda(s_1, 0) = a = \lambda(s_1, 1)$ and $0 \neq 1$).

Example 3.27. *It can be proven that the transducer M of Example 3.3 is also injective with delay 1, and is not injective with delay 0.*

In Chapter 5 we will see an efficient method for checking if a linear finite transducer is injective with delay τ, for any $\tau \in \mathbb{N}_0$.

Example 3.28. *The transducer* $M = \langle \{0,1\}, \{a,b\}, \{s_1, s_2\}, \delta, \lambda \rangle$ *induced by the diagram*



is not injective with delay 1 since, for example, $\lambda(s_1, 01) = \lambda(s_1, 11)$ *and* $0 \neq 1$.

It is obvious that, if a finite transducer M is injective with some delay $\tau \in \mathbb{N}_0$, then M is also injective with delay k, for $k \geq \tau$, which implies that it is also ω-injective. Tao [Tao09, Corollary 1.4.3] proved the following result, which shows that the converse is also true.

Theorem 3.29. *Let* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *be a finite transducer. If* M *is* ω*-injective, then there exists a non-negative integer* $\tau \leq \frac{|S|(|S|-1)}{2}$ *such that* M *is injective with delay* τ.

Example 3.30. *From the previous theorem we may conclude that the transducer* M *defined in Example 3.28 is not* ω*-injective, since it is not injective with delay* 1 *and the set of states has size 2.*

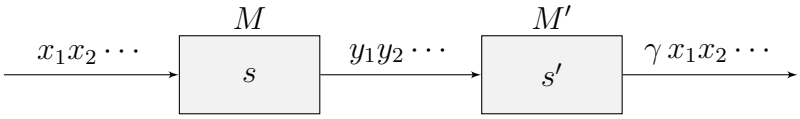
Since every ω-injective finite transducer is injective with some delay τ, our study of injectivity, presented in the following chapters, is confined to these latter transducers.

Naturally, injective transducers should have inverses of some sort. In order to describe the appropriate concept we introduce a notion of an inverse state of a given state.

Definition 3.31. *Let* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *and* $M' = \langle \mathcal{Y}, \mathcal{X}, S', \delta', \lambda' \rangle$ *be two finite transducers. Let* $s \in S$ *and* $s' \in S'$. *We say that* s' inverts s with delay τ *or* s' *is an* inverse state with delay τ *of* s *when*

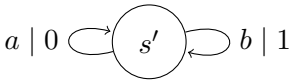
$$\forall \alpha \in \mathcal{X}^\omega,\ \lambda'(s', \lambda(s, \alpha)) = \gamma\alpha,\ \textit{for some } \gamma \in \mathcal{X}^\tau.$$

The figure below gives a schematic representation of this concept with $x_1x_2\cdots = \alpha$ and $y_1y_2\cdots = \lambda(s,\alpha)$.



Remark 3.32. *In the previous definition one may replace $\mathcal{X}^\omega$ by $\mathcal{X}^\star$, but then one should also replace $\lambda'(s',\lambda(s,\alpha)) = \gamma\alpha$ by $\lambda'(s',\lambda(s,\alpha)) = \gamma\alpha'$, where α' consists of the first $|\alpha| - \tau$ characters of α.*

Example 3.33. *Let $M' = \langle\{a,b\},\{0,1\},\{s'\},\delta',\lambda'\rangle$ be the finite transducer induced by the following diagram:*



We will see that the state s' of M' inverts the states s_1 and s_2 of M with delay 1, where M is the transducer defined in Example 3.2.

To prove that, it is enough to show that for all $x_1x_2 \in \{0,1\}^2$, and for all $s \in \{s_1,s_2\}$, one has

$$\lambda'(s',\lambda(s,x_1x_2)) = xx_1, \text{ for some } x \in \{0,1\}, \tag{3.2}$$

because this implies that for all $\alpha \in \{0,1\}^\omega$, and for all $s \in \{s_1,s_2\}$,

$$\lambda'(s',\lambda(s,\alpha)) = x\alpha, \text{ for some } x \in \{0,1\}.$$

Using the diagrams of the transducers one easily gets

$$\begin{aligned}
&\lambda'(s',\lambda(s_1,00)) = \lambda'(s',aa) = 00, && \lambda'(s',\lambda(s_1,10)) = \lambda'(s',ab) = 01,\\
&\lambda'(s',\lambda(s_1,01)) = \lambda'(s',aa) = 00, && \lambda'(s',\lambda(s_1,11)) = \lambda'(s',ab) = 01,\\
&\lambda'(s',\lambda(s_2,00)) = \lambda'(s',ba) = 10, && \lambda'(s',\lambda(s_2,10)) = \lambda'(s',bb) = 11,
\end{aligned}$$

$$\lambda'(s', \lambda(s_2, 01)) = \lambda'(s', ba) = 10, \qquad \lambda'(s', \lambda(s_2, 11)) = \lambda'(s', bb) = 11.$$

This proves that (3.2) *holds.*

Definition 3.34. *Let* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *be a finite transducer. One says that* M *is* left invertible with delay τ *if there is a transducer* $M' = \langle \mathcal{Y}, \mathcal{X}, S', \delta', \lambda' \rangle$ *such that*

$$\forall\, s \in S,\ \exists\, s' \in S',\ s' \text{ inverts } s \text{ with delay } \tau.$$

The transducer M' *is called a* left inverse with delay τ *of* M*.*

It is clear that, in the previous example, the transducer M' is a left inverse with delay 1 of M.

If M' is a left inverse with delay τ of M, then M' can recover the input of M with a delay of τ input symbols.

The following result establishes the fundamental relation between the injectivity of a transducer and the existence of a left inverse.

Theorem 3.35. *A finite transducer* $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ *is injective with delay* τ *if and only if there exists a finite transducer* $M' = \langle \mathcal{Y}, \mathcal{X}, S', \delta', \lambda' \rangle$ *such that* M' *is a left inverse with delay* τ *of* M*.*

Proof. The necessary condition is proven by Tao [Tao09, Theorem 1.4.4]. To prove the sufficient condition, assume that there is a transducer M' which is a left inverse with delay τ of M, for $\tau \in \mathbb{N}_0$. Let $s \in S$, $x, x' \in \mathcal{X}$, and $\alpha, \alpha' \in \mathcal{X}^\tau$. Then there is a state $s' \in S'$ such that

$$\lambda(s, x\alpha) = \lambda(s, x'\alpha') \implies \lambda'(s', \lambda(s, x\alpha)) = \lambda'(s', \lambda(s, x'\alpha')) \implies x = x'.$$

Therefore, M is injective with delay τ. □

3.1.2 Finite Transducers with Memory

Let A be a non-empty set and $j \in \mathbb{N}$. Define $\sigma_j : A^j \times A \to A^j$ by:

$$\sigma_j((a_1, \ldots, a_j), a) = (a_2, \ldots, a_j, a).$$

Definition 3.36. *Let $\phi : \mathcal{X}^{h+1} \times \mathcal{Y}^k \longrightarrow \mathcal{Y}$, with $h, k \in \mathbb{N}_0$ not simultaneously null, and $\mathcal{X}, \mathcal{Y}$ two non-empty finite sets. Let $M_\phi = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h \times \mathcal{Y}^k, \delta_\phi, \lambda_\phi \rangle$ be the finite transducer such that, for all $x \in \mathcal{X}$, $\alpha \in \mathcal{X}^h$, $\beta \in \mathcal{Y}^k$, and the state transition and output functions are given by:*

$$\begin{aligned} \delta_\phi(< \alpha, \beta >, x) &=< \sigma_h(\alpha, x), \sigma_k(\beta, y) >, \\ \lambda_\phi(< \alpha, \beta >, x) &= y, \end{aligned}$$

where $y = \phi(\alpha, x, \beta)$ and $< \ldots >$ is used to denote the states of this transducer. M_ϕ is called the finite transducer with memory (h, k) defined by ϕ. *If $k = 0$, then M_ϕ is said to be a* finite transducer with input memory $(h, 0)$.

As the name suggests, a finite transducer with memory is completely defined by its memory (h, k) and by the function ϕ. Notice that λ_ϕ and δ_ϕ are explicitly given by ϕ. Below, there is a schematic representation of the state transition function for this kind of transducers, where $x_1, \ldots, x_h, x \in \mathcal{X}$ and $y_1, \ldots, y_k, y \in \mathcal{Y}$.

$$< x_1, x_2, \ldots, x_h, y_1, y_2, \ldots, y_k > \xrightarrow{x \mid y} < x_2, \ldots, x_h, x, y_2, \ldots, y_k, y >$$

Example 3.37. *Let M_ϕ be the finite transducer with memory of order $(2, 1)$ defined by the map $\phi : \mathbb{F}_2^4 \to \mathbb{F}_2$ with $\phi(a, b, c, d) = c + b\,d$. Then $M_\phi = \langle \mathbb{F}_2, \mathbb{F}_2, \mathbb{F}_2^3, \delta_\phi, \lambda_\phi \rangle$ is such that*

$$\begin{aligned} &\lambda_\phi(< x_1, x_2, y_1 >, x) = \phi(x_1, x_2, x, y_1), \text{ and} \\ &\delta_\phi(< x_1, x_2, y_1 >, x) =< x_2, x, \lambda_\phi(< x_1, x_2, y_1 >, x) > . \end{aligned}$$

Take $s =< 1,1,1 > \in \mathbb{F}_2^3$. *Then,*

$$\lambda_\phi(s,0) = \phi(1,1,0,1) = 1, \text{ and}$$
$$\delta_\phi(s,0) =< 1,0,1 > .$$

Usually, finite transducers with memory of order (h,k) are defined by the set of equations

$$y_t = \phi(x_{t-h}, \ldots, x_{t-1}, x_t, y_{t-k}, \ldots, y_{t-1}), \text{ for } t \geq 0,$$

starting with some initial state to which one assigns negative indices. For example, the transducer in the previous example could be defined as follows. Let $M_\phi = \langle \mathbb{F}_2, \mathbb{F}_2, \mathbb{F}_2^3, \delta_\phi, \lambda_\phi \rangle$ be the finite transducer with memory of order $(2,1)$ defined by

$$y_t = x_t + x_{t-1}\, y_{t-1}, \text{ for } t \geq 0,$$

where $s =< x_{-2}, x_{-1}, y_{-1} >$ is the initial state of the transducer. With this kind of notation we are assuming that

$$y_0 y_1 \cdots = \lambda_\phi(< x_{-2}, x_{-1}, y_{-1} >, x_0 x_1 \cdots).$$

where $x_i \in \mathbb{F}_2$, for $i \geq -2$, and $y_j \in \mathbb{F}_2$, for $j \geq -1$.

Example 3.38. *Let* $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, (\mathbb{F}_2^2)^2 \times \mathbb{F}_2^3, \delta, \lambda \rangle$ *be the finite transducer with memory of order* $(2,1)$ *defined by*

$$y_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} x_{t-2} + y_{t-1}, \text{ for } t \geq 0,$$

where $x_i \in \mathbb{F}_2^2$, *for* $i \geq -2$, $y_j \in \mathbb{F}_2^3$, *for* $j \geq -1$, *and* $< x_{-2}, x_{-1}, y_{-1} >$ *is the initial state of the transducer.*

Take $x_{-2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $x_{-1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $y_{-1} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, *and* $s =< x_{-2}, x_{-1}, y_{-1} >$. *Then, for example,*

$$\lambda\left(s, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

If, in the definition of finite transducer with memory, $(\mathcal{Y}, +)$ is a group (not necessarily Abelian) and the function ϕ is of the form

$$\phi = f(x_1, x_2, \ldots, x_h, x_{h+1}) + g(y_1, y_2, \ldots, y_k),$$

for some $f : \mathcal{X}^{h+1} \to \mathcal{Y}$ and $g : \mathcal{Y}^k \to \mathcal{Y}$, one says that M_ϕ is a *separable finite transducer with memory*, denoted by $M_{f,g}$. Notice that, in particular, a finite transducer with input memory $(h, 0)$ is a separable finite transducer.

Example 3.39. *The transducer defined in the previous example is a separable finite transducer, while the transducer defined in Example 3.37 is not separable.*

The following result about separable finite transducers is mentioned by Zongduo *et al.* [ZDL98] without proof.

Theorem 3.40. *Let* $\mathcal{Y}$ *be a group, denoted additively. Then the separable transducer* $M_{f,g} = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h \times \mathcal{Y}^k, \delta_{f,g}, \lambda_{f,g} \rangle$ *is injective with delay* τ *if and only if the transducer* $M_f = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h, \delta_f, \lambda_f \rangle$ *is injective with delay* τ.

Proof. Notice that, given $s_1 \in \mathcal{X}^h$, $s_2 \in \mathcal{Y}^k$, $x \in \mathcal{X}$, one can write

$$\lambda_{f,g}(< s_1, s_2 >, x) = f(s_1, x) + g(s_2). \tag{3.3}$$

Also, if $s_1 \in \mathcal{X}^h$, $s_2 \in \mathcal{Y}^k$, $x \in \mathcal{X}$, and $\alpha \in \mathcal{X}^\tau$, then $\lambda_{f,g}(< s_1, s_2 >, x\alpha)$ is just a

sequence of elements as in (3.3). Since, obviously,

$$f(s_1, x) + g(s_2) = f(s_1, x') + g(s_2) \iff f(s_1, x) = f(s_1, x'),$$

for all $s_1 \in \mathcal{X}^h$, $s_2 \in \mathcal{Y}^k$, $x, x' \in \mathcal{X}$, and $\alpha, \alpha' \in \mathcal{X}^\tau$, one concludes that

$$\lambda_{f,g}(< s_1, s_2 >, x\alpha) = \lambda_{f,g}(< s_1, s_2 >, x'\alpha')$$

is equivalent to

$$\lambda_f(< s_1 >, x\alpha) = \lambda_f(< s_1 >, x'\alpha').$$

From this, the claim made follows immediately. □

3.2 The Notion of Linear Finite Transducer

Definition 3.41. *If $\mathcal{X}, \mathcal{Y}$ and S are vector spaces over a field $\mathbb{F}$, and both $\delta : S \times \mathcal{X} \to S$ and $\lambda : S \times \mathcal{X} \to Y$ are linear maps, then $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ is called a* linear finite transducer (LFT) *over $\mathbb{F}$, and we say that the* size *of M, denoted* size(M)*, is the dimension of S as a vector space.*

Example 3.42. *Let $M = \langle \mathbb{F}_2^3, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta, \lambda \rangle$ be the transducer defined by:*

$$\delta(s, x) = (s_2 + x_1, s_1 + x_2 + x_3),$$
$$\lambda(s, x) = (s_1 + x_1 + x_3, s_2 + x_2),$$

for all $s = (s_1, s_2) \in \mathbb{F}_2^2$, and for all $x = (x_1, x_2, x_3) \in \mathbb{F}_2^3$. The state transition function $\delta : \mathbb{F}_2^5 \to \mathbb{F}_2^2$ and the output function $\lambda : \mathbb{F}_2^5 \to \mathbb{F}_2^2$ are linear maps, therefore, M is an LFT over $\mathbb{F}_2$ and the size of M is $\dim(\mathbb{F}_2^2) = 2$. Moreover, if one considers the standard bases of $\mathbb{F}_2^5$ and $\mathbb{F}_2^2$, those maps are represented in terms of matrices in the

following way

$$\begin{aligned}
\delta(s,x) &= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_1 & s_2 & x_1 & x_2 & x_3 \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} s + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} x,
\end{aligned}$$

$$\begin{aligned}
\lambda(s,x) &= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 & s_2 & x_1 & x_2 & x_3 \end{bmatrix}^{\mathrm{T}} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} s + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} x.
\end{aligned}$$

Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be an LFT over a field $\mathbb{F}$. If $\mathcal{X}, \mathcal{Y}$, and S have dimensions ℓ, m and n, respectively, then there exist matrices $A \in \mathcal{M}_n(\mathbb{F})$, $B \in \mathcal{M}_{n\times\ell}(\mathbb{F})$, $C \in \mathcal{M}_{m\times n}(\mathbb{F})$, and $D \in \mathcal{M}_{m\times\ell}(\mathbb{F})$, such that, in the appropriate bases,

$$\begin{aligned}
\delta(s,x) &= As + Bx, \\
\lambda(s,x) &= Cs + Dx,
\end{aligned}$$

for all $s \in S$, $x \in \mathcal{X}$. From the computations made on the previous example it is easy to understand how the matrices can be constructed from the maps δ and γ. The matrices A, B, C, D are called the *structural matrices* of M, and ℓ, m, n are called its *structural parameters*.

Sometimes we define the LFT by the quadruple (A, B, C, D), where A, B, C, D are its structural matrices.

Notice that if M_1 and M_2 are two equivalent LFTs with structural parameters ℓ_1, m_1, n_1 and ℓ_2, m_2, n_2, respectively, then, from the definition of equivalent transducers, one has $\ell_1 = \ell_2$ and $m_1 = m_2$.

An LFT such that C is the null matrix (with the adequate dimensions) is called *trivial*.

Let $\mathcal{L}$ denote the set of LFTs over a given field $\mathbb{F}$, and $\mathcal{L}_n$ the set of the transducers in $\mathcal{L}$ with size n. The restriction to $\mathcal{L}$ of the relation $\sim$ of FTs equivalence is also represented by $\sim$, and the restriction to $\mathcal{L}_n$ is denoted by $\sim_n$. The notation $\mathcal{L}_{\ell,m,n}$ is used to represent the set of LFTs with structural parameters ℓ, m, n.

Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be an LFT over a field $\mathbb{F}$ with structural matrices A, B, C, D. Starting at a state s_0 and reading an input sequence $x_0x_1x_2\cdots$, one gets a sequence of states $s_0s_1s_2\cdots$ and a sequence of outputs $y_0y_1y_2\cdots$ satisfying the relations

$$\begin{aligned} s_{t+1} &= \delta(s_t, x_t) = As_t + Bx_t, \\ y_t &= \lambda(s_t, x_t) = Cs_t + Dx_t, \end{aligned}$$

for all $t \geq 0$. The following result can be proven by induction [Tao09, Theorem 1.3.1]. Nonetheless, in Chapter 5.1, we present a more conceptual proof using formal power series.

Theorem 3.43. *For an LFT as above,*

$$s_i = A^i s_0 + \sum_{j=0}^{i-1} A^{i-j-1} B x_j, \tag{3.4}$$

and

$$y_i = CA^i s_0 + \sum_{j=0}^{i} H_{i-j} x_j, \tag{3.5}$$

for $i \in \mathbb{N}_0$, where $H_0 = D$, and $H_j = CA^{j-1}B$, $j > 0$.

One can associate to an LFT a family of matrices which are very important in the study of its equivalence class, as will be clear throughout Chapter 4.

Definition 3.44. *Let $M = (A, B, C, D)$ be an LFT of size $n \in \mathbb{N}$. The matrix*

$$\Delta_M^{(k)} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix}$$

*is called the k-*diagnostic matrix *of M, where $k \in \mathbb{N} \cup \{\infty\}$. The matrix $\Delta_M^{(n)}$ will be simply denoted by Δ_M and will be referred to as the* diagnostic matrix *of M. The matrix $\Delta_M^{(2n)}$ will be denoted by $\hat{\Delta}_M$ and called the* augmented diagnostic matrix *of M.*

3.3 Equivalence of Sates and of LFTs

Tao, in his book, presents the following necessary and sufficient condition for the equivalence of two states of LFTs [Tao09, Theorem 1.3.3].

Theorem 3.45. *Let $M_1 = \langle \mathcal{X}, \mathcal{Y}_1, S_1, \delta_1, \lambda_1 \rangle$ and $M_2 = \langle \mathcal{X}, \mathcal{Y}_2, S_2, \delta_2, \lambda_2 \rangle$ be two LFTs. Let $s_1 \in S_1$, and $s_2 \in S_2$. Then, $s_1 \sim s_2$ if and only if the null states of M_1 and M_2 are equivalent and $\lambda_1(s_1, 0^\omega) = \lambda_2(s_2, 0^\omega)$.*

And, as a consequence, he also presents a necessary and sufficient condition for the equivalence of two LFTs [Tao09, Theorem 1.3.3].

Corollary 3.46. *Let M_1 and M_2 be two LFTs. Then, $M_1 \sim M_2$ if and only if their null states are equivalent and $\{\lambda_1(s_1, 0^\omega) \mid s_1 \in S_1\} = \{\lambda_2(s_2, 0^\omega) \mid s_2 \in S_2\}$.*

In this section, we explain how these conditions can be easily checked using linear algebra, providing a result which is essential in Subsection 4.2 to compute the sizes of equivalence classes in $\mathcal{L}_n/\!\sim_n$.

For the remainder of this section, let $M_1 = \langle \mathcal{X}, \mathcal{Y}_1, S_1, \delta_1, \lambda_1 \rangle$, $M_2 = \langle \mathcal{X}, \mathcal{Y}_2, S_2, \delta_2, \lambda_2 \rangle$ be two LFTs with structural matrices A_1, B_1, C_1, D_1, and A_2, B_2, C_2, D_2 respectively. Let $n_1 = \text{size}(M_1)$ and $n_2 = \text{size}(M_2)$. To simplify the notation, let $\tilde{\Delta}_1 = \Delta_{M_1}^{(n_1+n_2)}$ and $\tilde{\Delta}_2 = \Delta_{M_2}^{(n_1+n_2)}$.

Lemma 3.47. *Let $s_1 \in S_1$ and $s_2 \in S_2$. Then, $\lambda_1(s_1, 0^\omega) = \lambda_2(s_2, 0^\omega)$ if and only if $\tilde{\Delta}_1 s_1 = \tilde{\Delta}_2 s_2$.*

Proof. From Theorem 3.43, one has that $\lambda_1(s_1, 0^\omega) = \lambda_2(s_2, 0^\omega)$ if and only if $C_1 A_1^i s_1 = C_2 A_2^i s_2$, for $i \geq 0$. Let p_1 be the characteristic polynomial of A_1, and p_2 the characteristic polynomial of A_2. Then, p_1 and p_2 are monic polynomials of order n_1 and n_2, respectively. Moreover, by the Cayley-Hamilton theorem, $p_1(A_1) = p_2(A_2) = 0$. Thus, $p = p_1 p_2$ is a monic polynomial of order $n_1 + n_2$ such that $p(A_1) = p(A_2) = 0$. Therefore $A_1^{n_1+n_2+k}$ and $A_2^{n_1+n_2+k}$, with $k \geq 0$, are linear combinations of lower powers of A_1 and A_2, respectively, with the same coefficients. Consequently, $C_1 A_1^i s_1 = C_2 A_2^i s_2$ for $i \geq 0$ is equivalent to $C_1 A_1^i s_1 = C_2 A_2^i s_2$ for $i = 0, 1, \ldots, n_1 + n_2 - 1$, and the result follows. □

Lemma 3.48. *The null states of M_1 and M_2 are equivalent if and only if*

$$D_1 = D_2 \text{ and } \tilde{\Delta}_1 B_1 = \tilde{\Delta}_2 B_2.$$

Proof. By definition, the null states of M_1 and M_2 are equivalent if and only if

$$\forall \alpha \in \mathcal{X}^\star,\ \lambda_1(0, \alpha) = \lambda_2(0, \alpha).$$

By Theorem 3.43, this is equivalent to:

$$\sum_{j=0}^{i} H_{i-j} x_j = \sum_{j=0}^{i} H'_{i-j} x_j,\ i = 0, 1, \ldots, |\alpha|,$$

where $\alpha = x_0 x_1 \cdots x_{|\alpha|} \in \mathcal{X}^\star$, $H_0 = D_1$, $H'_0 = D_2$ and $H_j = C_1 A_1^{j-1} B_1$, $H'_j = C_2 A_2^{j-1} B_2$, for $j > 0$. That is, $\forall x_0, x_1, \ldots, x_{|\alpha|} \in \mathcal{X}$ the following equations are

simultaneously satisfied:

$$\begin{aligned} D_1x_0 &= D_2x_0 \\ D_1x_1 + C_1B_1x_0 &= D_2x_1 + C_2B_2x_0 \\ D_1x_2 + C_1B_1x_1 + C_1A_1B_1x_0 &= D_2x_2 + C_2B_2x_1 + C_2A_2B_2x_0 \\ &\vdots \\ D_1x_{|\alpha|} + \cdots + C_1A_1^{(|\alpha|-1)}B_1x_0 &= D_2x_{|\alpha|} + \cdots + C_2A_2^{(|\alpha|-1)}B_2x_0. \end{aligned}$$

Using the characteristic polynomials of A_1 and A_2, as in the proof of Lemma 3.47, one sees that when $|\alpha| \geq u$ the equations after the first u of them are implied by the previous ones. From the arbitrariness of α, it then follows that the system is satisfied if and only if

$$D_1 = D_2 \quad \text{and} \quad \tilde{\Delta}_1B_1 = \tilde{\Delta}_2B_2.$$

□

The next result states that the $(n_1 + n_2)$-diagnostic matrices of two LFTs, of sizes n_1 and n_2, can be used to verify if two of their states are equivalent. It follows directly from Theorem 3.45 and from the previous two lemmas.

Theorem 3.49. *Let $s_1 \in S_1$ and $s_2 \in S_2$. Then $s_1 \sim s_2$ if and only if the following two conditions are simultaneously satisfied:*

1. $\tilde{\Delta}_1 s_1 = \tilde{\Delta}_2 s_2$
2. $D_1 = D_2$ *and* $\tilde{\Delta}_1B_1 = \tilde{\Delta}_2B_2$.

Corollary 3.50. *Let $s_1 \in S_1$ and $s_2 \in S_2$. If $M_1 \sim M_2$, then $s_1 \sim s_2$ if and only if $\tilde{\Delta}_1 s_1 = \tilde{\Delta}_2 s_2$.*

Proof. From Corollary 3.46, if $M_1 \sim M_2$ then the null states of M_1 and M_2 are equivalent, that is, $D_1 = D_2$ and $\tilde{\Delta}_1B_1 = \tilde{\Delta}_2B_2$. The result then follows. □

Corollary 3.51. *Let M be an LFT, and $s_1, s_2 \in M$. Then, $s_1 \sim s_2$ if and only if $\Delta_M s_1 = \Delta_M s_2$.*

Proof. From the last Corollary, $s_1 \sim s_2$ if and only if $\hat{\Delta}_M s_1 = \hat{\Delta}_M s_2$, that is, if and only if $CA^i s_1 = CA^i s_2$, for $i = 0, 1, \ldots, 2n-1$. Since the minimal polynomial of A has, at most, degree n, this latter condition is equivalent to $CA^i s_1 = CA^i s_2$, for $i = 0, 1, \ldots, n-1$. Thus, $s_1 \sim s_2$ if and only if $\Delta_M s_1 = \Delta_M s_2$. □

Example 3.52. *Using the previous corollary it is quite easy to verify that the states $s_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $s_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ of the transducer $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta, \lambda \rangle$ defined in Example 3.10 are equivalent. Recall that the structural matrices of M are*

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ C = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Then

$$\Delta_M = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

and

$$\Delta_M s_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \Delta_M s_2.$$

Therefore $s_1 \sim s_2$.

Corollary 3.53. *Let M be an LFT over a field $\mathbb{F}$. Then, M is minimal if and only if* $\operatorname{rank}(\Delta_M) = \operatorname{size}(M)$.

Proof. Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be an LFT over a field $\mathbb{F}$. It is enough to see that the linear application $\varphi : S/\!\sim \to \mathbb{F}^{nm}$ defined by $\varphi([\, s \,]_\sim) = \Delta_M s$ is well-defined and

injective. Let $[s_1], [s_2] \in S/\sim$. From Corollary 3.51, one gets

$$[s_1] = [s_2] \Leftrightarrow s_1 \sim s_2 \Leftrightarrow \Delta_M s_1 = \Delta_M s_2 \Leftrightarrow \varphi([s_1]) = \varphi([s_2])$$

Therefore, φ is well-defined and injective. □

Lemma 3.54. *Let $M \in \mathcal{L}_n$ with structural matrices A, B, C, D. Then,*

$$\operatorname{rank}(\Delta_M^{(k)}) = \operatorname{rank}(\Delta_M),\ \forall k \geq n.$$

Proof. The degree of the minimal polynomial of A is at most n, and so the matrices CA^k, for $k \geq n$, are linear combinations of $C, CA^1, \cdots, CA^{n-1}$. □

The following theorem gives a pair of conditions that have to be satisfied for two LFTs to be equivalent.

Theorem 3.55. *For LFTs M_1 and M_2 as above, $M_1 \sim M_2$ if and only if the following two conditions are simultaneously verified:*

1. $\operatorname{rank}(\tilde{\Delta}_1) = \operatorname{rank}([\tilde{\Delta}_1 \mid \tilde{\Delta}_2]) = \operatorname{rank}(\tilde{\Delta}_2)$;
2. $D_1 = D_2$ *and* $\tilde{\Delta}_1 B_1 = \tilde{\Delta}_2 B_2$.

Proof. From Corollary 3.46 one has that $M_1 \sim M_2$ if and only if the null states of M_1 and M_2 are equivalent, and $\{\lambda_1(s_1, 0^\omega) \mid s_1 \in S_1\} = \{\lambda_2(s_2, 0^\omega) \mid s_2 \in S_2\}$.

From Lemma 3.48 we already know that the null states are equivalent if and only if

$$D_1 = D_2 \text{ and } \tilde{\Delta}_1 B_1 = \tilde{\Delta}_2 B_2.$$

From Lemma 3.47, one has that

$$\{\lambda_1(s_1, 0^\omega) \mid s_1 \in S_1\} = \{\lambda_2(s_2, 0^\omega) \mid s_2 \in S_2\}$$

if and only if

$$\{\tilde{\Delta}_1 s_1 \mid s_1 \in S_1\} = \{\tilde{\Delta}_2 s_2 \mid s_2 \in S_2\}.$$

This means that the column space of $\tilde{\Delta}_1$ is equal to the column space of $\tilde{\Delta}_2$, which is true if and only if there exist matrices X, Y such that $\tilde{\Delta}_2 = \tilde{\Delta}_1 X$ and $\tilde{\Delta}_1 = \tilde{\Delta}_2 Y$. But, from Lemma 2.18, this happens if and only if $\text{rank}(\tilde{\Delta}_1) = \text{rank}([\tilde{\Delta}_1 \mid \tilde{\Delta}_2])$ and $\text{rank}(\tilde{\Delta}_2) = \text{rank}([\tilde{\Delta}_1 \mid \tilde{\Delta}_2])$. □

Example 3.56. *Let* $M_1 = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, \mathbb{F}_2^2, \delta_1, \lambda_1 \rangle$ *be the LFT defined by the following structural matrices*

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ B_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},\ C_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},\ D_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix},$$

and let $M_2 = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, \mathbb{F}_2, \delta_2, \lambda_2 \rangle$ *be the LFT defined by the matrices*

$$A_2 = \begin{bmatrix} 1 \end{bmatrix},\ B_2 = \begin{bmatrix} 1 & 0 \end{bmatrix},\ C_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ D_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Notice that $\text{size}(M_1) = 2$ *and* $\text{size}(M_2) = 1$. *Using the previous results we will prove the following claims:*

1. *The states* $s_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ *of* M_1 *and* $s_2 = \begin{bmatrix} 1 \end{bmatrix}$ *of* M_2 *are equivalent.*

2. M_1 *is not minimal and* M_2 *is minimal.*

3. $M_1 \sim M_2$.

From the structural matrices of M and M' one gets that

$$\tilde{\Delta}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \textit{and} \quad \tilde{\Delta}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Therefore

$$\tilde{\Delta}_1 s_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \tilde{\Delta}_2 s_2.$$

Moreover

$$\tilde{\Delta}_1 B_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} = \tilde{\Delta}_2 B_2.$$

Since $D_1 = D_2$ it follows, from Theorem 3.49, that $s_1 \sim s_2$.

To prove the second claim we just have to notice that $\mathrm{rank}(\Delta_{M_1}) = 1 < 2 = size(M_1)$ *and* $\mathrm{rank}(\Delta_{M_2}) = 1 = \mathrm{size}(M_2)$. *Thus, by Corollary 3.53, M_1 is not minimal and M_2 is minimal.*

Finally, one has

$$\mathrm{rank}([\tilde{\Delta}_1 \mid \tilde{\Delta}_2]) = \mathrm{rank}\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}\right) = 1,$$

then $\mathrm{rank}(\tilde{\Delta}_1) = \mathrm{rank}([\tilde{\Delta}_1 \mid \tilde{\Delta}_2]) = \mathrm{rank}(\tilde{\Delta}_2)$ *and, from Theorem 3.55, $M_1 \sim M_2$.*

It is important to recall, at this moment, that the size of an LFT is the only structural parameter that can vary between transducers of the same equivalence class in $\mathcal{L}/\sim$. Moreover, the size of an LFT of an equivalence class $[M]_\sim$, can never be smaller than $\mathrm{rank}(\Delta_{M'})$, where M' is a minimal transducer in $[M]_\sim$.

The following result shows that every LFT of size n_1 equivalent to M_1 has an augmented diagnostic matrix of the form $\hat{\Delta}_{M_1}X$, for some invertible matrix X in $\mathcal{M}_{n_1}$. It is a direct consequence of Lemma 2.19 and of the first point of Theorem 3.55.

Corollary 3.57. *If $n = n_1 = n_2$, $S_1 = S_2$, and $M_1 \sim M_2$, then there is an invertible matrix $X \in \mathcal{M}_n$ such that $\hat{\Delta}_{M_2} = \hat{\Delta}_{M_1}X$.*

3.4 Minimisation

In this section we give a method to obtain a minimal LFT equivalent to a given LFT.

Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a linear finite transducer over $\mathbb{F}_q$ of size n. Consider the diagnostic matrix of M, Δ_M. From Corollary 3.51, one knows that two states s_1 and s_2 of M are equivalent if and only if $\Delta_M s_1 = \Delta_M s_2$. Moreover, from Corollary 3.53 one also knows that M is minimal if and only if $\text{rank}(\Delta_M) = n$. Assume that M is not minimal. Let K be a matrix consisting of $\text{rank}(\Delta_M)$ linearly independent rows of Δ_M. Then, K is right invertible and two states, s_1 and s_2, of M are equivalent if and only if $Ks_1 = Ks_2$. Let R be a right inverse of K and $S' = \{Ks \mid s \in S\}$. Notice that $S' = \mathbb{F}_q^{\text{rank}(K)}$ and, therefore, is a vector space of dimension $\text{rank}(K)$. Let $M' = \langle \mathcal{X}, \mathcal{Y}, S', \delta', \lambda' \rangle$ be the LFT defined by the structural matrices

$$A' = KAR, \qquad B' = KB,$$
$$C' = CR, \qquad D' = D.$$

Theorem 3.58. *M' as before is minimal and equivalent to M.*

Proof. To prove the theorem, we show that M' and $M/\!\sim$ are isomorphic. Consider the mapping ψ defined as follows.

$$\begin{array}{rccc} \psi: & S/\!\sim & \longrightarrow & S' \\ & [s] & \longmapsto & Ks \end{array}$$

It is enough to prove that ψ is well defined and bijective, since, from Theorems 3.22 and 3.17, $M/\!\sim$ is minimal and equivalent to M. To prove that ψ is well defined and injective, let $[s_1]$, $[s_2] \in S/\!\sim$. Then, one has

$$[s_1] = [s_2] \;\Leftrightarrow\; s_1 \sim s_2 \;\Leftrightarrow\; Ks_1 = Ks_2 \;\Leftrightarrow\; \psi([s_1]) = \psi([s_2]).$$

The surjectiveness of ψ follows immediately from the fact that K is right invertible. □

Given a non-minimal LFT M, the previous discussion gives an algorithm to minimise M, namely:

1. Determine Δ_M and $\text{rank}(\Delta_M)$.

2. Construct a submatrix K of Δ_M consisting of $\text{rank}(\Delta_M)$ rows linearly independent.

3. Compute a right inverse of K, R.

4. Compute the structural matrices, A', B', C', D', of a minimal transducer equivalent to M:

$$A' = KAR, \qquad B' = KB,$$
$$C' = CR, \qquad D' = D.$$

Example 3.59. *Let $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta, \lambda \rangle$ be the LFT over $\mathbb{F}_2$ defined by the following structural matrices:*

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},\ C = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Let us find a minimal LFT equivalent to M using the process described above.

1. *One has*
$$\Delta_M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},$$
and $\text{rank}(\Delta_M) = 1$ *(which implies that the transducer is not minimal).*

2. *Let* $K = \begin{bmatrix} 0 & 1 \end{bmatrix}$. *$K$ is a submatrix of M formed by* $\text{rank}(M) = 1$ *row.*

3. *A right inverse of K is* $R = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

4. *The linear transducer $M' = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, \mathbb{F}_2, \delta', \lambda' \rangle$ over $\mathbb{F}_2$ defined by the structural*

matrices

$$A' = KAR = [1], \qquad B' = KB = \begin{bmatrix} 1 & 1 \end{bmatrix},$$
$$C' = CR = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad D' = D = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

is minimal and equivalent to M.

Chapter 4

Size and Number of Equivalence Classes of LFTs

From now on, we consider only LFTs defined over finite fields with q elements, $\mathbb{F}_q$, because these are the ones fitted for cryptographic uses.

The isomorphisms between LFTs that shall be considered below are always linear isomorphisms. Let M, N be two LFTs. If there is a linear isomorphism between M and N, we write $M \simeq_l N$, instead of simply $M \simeq N$.

4.1 Canonical Linear Finite Transducers

In this section, for each equivalence class in $\mathcal{L}/\sim$, we single out an LFT for which we can give a complete characterisation. Using this characterisation we establish a notion of *canonical* LFT.

Proposition 4.1. *Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a non-trivial LFT over $\mathbb{F}_q$ of size $n \in \mathbb{N}$, with structural matrices A, B, C and D. Let $X \in GL_n(\mathbb{F}_q)$, and $M_X = \langle \mathcal{X}, \mathcal{Y}, S, \delta_X, \lambda_X \rangle$ be the LFT defined by the structural matrices*

$$A_X = X^{-1}AX, \qquad B_X = X^{-1}B,$$
$$C_X = CX, \qquad D_X = D.$$

Then, $M_X \simeq_l M$ *and* $\hat{\Delta}_{M_X} = \hat{\Delta}_M X$. *Conversely, given* $N \in \mathcal{L}_n$ *such that* $N \simeq_l M$, *then*

$$\exists X \ \in GL_n(\mathbb{F}_q) : N = M_X.$$

Proof. Let $\psi : S \to S$ be the bijective linear map defined by $\psi(s) = X^{-1}s$. Then $M_X = M_\psi$, where M_ψ is the transducer constructed from M and ψ as explained in Remark 3.7. Therefore, $M_X \simeq_l M$. Proving that $\hat{\Delta}_{M_X} = \hat{\Delta}_M X$ is also quite easy:

$$\hat{\Delta}_M X = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{2n-1} \end{bmatrix} X = \begin{bmatrix} CX \\ CAX \\ \vdots \\ CA^{2n-1}X \end{bmatrix} = \hat{\Delta}_{M_X}.$$

Now, assume that $N = \langle \mathcal{X}, \mathcal{Y}, S, \delta_N, \lambda_N \rangle \in \mathcal{L}_n$ is such that $N \simeq_l M$. Then, there is a linear isomorphism $\varphi : S \to S$ which satisfy conditions mentioned in Definition 3.6. Let P be the matrix of φ relative to any basis, and A_N, B_N, C_N, D_N, the structural matrices of N on that basis of S. Then, $P \in GL_n(\mathbb{F}_q)$, and we will see that $N = M_P$, where M_P is the LFT constructed from M and P as defined in the proposition. Let $x = 0_{\mathcal{X}}$ and $s \in S$. From the first condition of the definition, one gets

$$\varphi(\delta(s,0)) = \delta_N(\varphi(s),0) \Leftrightarrow PAs = A_N Ps \Leftrightarrow (PA - A_N P)s = 0.$$

From the arbitrariness of s, this is equivalent to $PA - A_N P = 0$. Since P is invertible, one gets $A_N = PAP^{-1} = A_P$. The second condition yields

$$\lambda(s,0) = \lambda_N(\varphi(s),0) \Leftrightarrow Cs = C_N Ps \Leftrightarrow (C - C_N P)s = 0.$$

Again, from the arbitrariness of s, this is equivalent to $C - C_N P = 0$. Thus, $C_N = CP^{-1} = C_P$.

Now, let $s = 0_S$ and $x \in \mathcal{X}$. Using a similar method, one gets $B_N = PB = B_P$ and $D_N = D = D_P$. Hence, $N = M_P$. □

Corollary 4.2. *In every non-trivial equivalence class for $\sim$ there is exactly one minimal LFT, M, such that Δ_M is in reduced column echelon form.*

Proof. Let M be a minimal LFT of size n. Let X be the invertible matrix such that $\Delta_M X$ is in reduced column echelon form. Let M_X be the LFT constructed from M and X as defined in Proposition 4.1. Then, $M_X \in [M]_\sim$ and $\Delta_{M_X} = \Delta_M X$ which is in reduced column echelon form. Notice that M_X is minimal because $M_X \simeq_l M$ and M is minimal. The uniqueness of such LFT follows from the fact that minimal LFTs that are equivalent are also isomorphic. □

Finally we can state the definition of canonical LFT here considered.

Definition 4.3. *Let M be a minimal LFT of size $n \in \mathbb{N}$. One says that M is a* canonical LFT *if Δ_M is in reduced column echelon form.*

Given M, an LFT, from the proofs of Proposition 4.1 and Corollary 4.2, one can easily identify and construct the canonical transducer in the equivalence class $[M]_\sim$. To do that we can follow the steps below.

1. Find a minimal transducer, $M_1 = (A_1, B_1, C_1, D_1)$, equivalent to M using, for example, the procedure presented in Section 3.4.

2. Determine Δ_{M_1} and find the invertible matrix X such that $\Delta_{M_1} X$ is in reduced column echelon form.

3. Determine X^{-1} and compute the structural matrices, A', B', C', D', of the

canonical transducer equivalent to M:

$$A' = X^{-1}A_1X, \qquad B' = X^{-1}B_1,$$
$$C' = C_1X, \qquad D' = D_1.$$

Example 4.4. *Let $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta, \lambda \rangle$ be the LFT over $\mathbb{F}_2$ defined by the following structural matrices*

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},\ C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

1. *Since*
$$\Delta_M = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$
and $\operatorname{rank}(\Delta_M) = 2 = \operatorname{size}(M)$*, the transducer M is minimal. Take $M_1 = M$.*

2. *The invertible matrix X such that $\Delta_M X$ is in reduced column echelon form is*
$$X = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

3. *Since $X^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, the structural matrices of the canonical transducer in $[M]_\sim$ are:*
$$A' = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ B' = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},\ C' = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \text{and } D' = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

4.2 Size of Equivalence Classes

In this section, we first give some important consequences of Proposition 4.1, and then we discuss how to compute the size of the equivalence classes in $\mathcal{L}_n/\sim_n$. The results presented, as well as the techniques in their proofs, allow us to enumerate the LFTs in $[M]_{\sim_n}$, where M is an LFT of size $n \in \mathbb{N}$.

Definition 4.5. *Let $M \in \mathcal{L}$ and $n = \text{size}(M)$. The set $\{\hat{\Delta}_{M'} \mid M' \in [M]_{\sim_n}\}$ will be called* the diagnostic set of M *and is denoted by D_M.*

Proposition 4.1 has the following interesting consequences.

i. From the structural matrices of an LFT of size $n \in \mathbb{N}$, A, B, C, D, one can enumerate the set of transducers M' such that $M' \simeq_l M$. Let $S_{\simeq_l M}$ denote that set. Then,

$$S_{\simeq_l M} = \{M_X \mid X \in GL_n(\mathbb{F}_q)\},$$

where M_X is the LFT constructed from M and X as in Proposition 4.1.

ii. Every matrix of the form $\hat{\Delta}_M X$, for $X \in GL_n(\mathbb{F}_q)$, is the augmented diagnostic matrix of at least a transducer in $[M]_{\sim_n}$. Moreover, from Corollary 3.57, one already knows that augmented diagnostic matrices of LFTs in $[M]_{\sim_n}$ are all of the form $\hat{\Delta}_M X$, for some X in $GL_n(\mathbb{F}_q)$. Therefore, the following equality holds.

$$D_M = \left\{\hat{\Delta}_M X \mid X \in GL_n(\mathbb{F}_q)\right\}. \tag{4.1}$$

iii. One knows that, if M_1 and M_2 are two minimal and equivalent LFTs with the same output alphabet, then M_1 and M_2 are isomorphic [Tao09, page 11]. Consequently, if M is minimal of size $n \in \mathbb{N}$, then $S_{\simeq_l M} = [M]_{\sim_n}$. Therefore, if X is a matrix in $GL_n(\mathbb{F}_q)$ and M is a minimal LFT in $\mathcal{L}_n$, then there is one and only one LFT, M', in $[M]_{\sim_n}$ such that

$$\hat{\Delta}_{M'} = \hat{\Delta}_M X.$$

The same is not true if M is not minimal as it will be seen later in this section.

iv. Given an LFT, M, of size $n \in \mathbb{N}$, a matrix X in $GL_n(\mathbb{F}_q)$, and setting $E_\Delta = \{M' \in [M]_{\sim_n} \mid \hat{\Delta}_{M'} = \Delta\}$, it is straightforward to see that the mapping

$$f_X : \begin{array}{ccc} E_{\Delta_M} & \longrightarrow & E_{\Delta_M X} \\ M & \longmapsto & M_X \end{array},$$

where M_X is the transducer constructed from M and X as defined in Proposition 4.1, is bijective. Therefore $|E_{\Delta_M}| = |E_{\Delta_M X}|$.

Notice that, from iv., any two matrices in D_M are associated to exactly the same number of transducers in $[M]_{\sim_n}$. Then, to obtain $|[M]_{\sim_n}|$, we just need to follow the two steps below.

1. Compute the size of the diagnostic set of M, *i.e*, $|D_M|$.

2. Choose a matrix in D_M and compute the number of LFTs in $[M]_{\sim_n}$ that are associated to it. Recall that D_M is the set of augmented diagnostic matrices of transducers in $[M]_{\sim_n}$.

In this way, the product of the two numbers thus obtained is $|[M]_{\sim_n}|$.

From ii., computing the size of D_M is equivalent to computing the number of distinct matrices of the form $\hat{\Delta}_M X$, where $X \in GL_n(\mathbb{F}_q)$. Consequently, from Theorem 2.20,

$$|D_M| = \prod_{i=0}^{\operatorname{rank}(\hat{\Delta}_M)-1} (q^n - q^i).$$

The key idea in step 2. is to choose an augmented diagnostic matrix that makes the computations easier. Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be an LFT with structural parameters $\ell, m, n \in \mathbb{N}$, and let $M_1 \in \mathcal{L}_{n_1}$ be a minimal LFT equivalent to M (one knows that M_1 exists from Section 3.4), where $n_1 = \operatorname{size}(M_1) = \operatorname{rank}(\Delta_M)$. Let A_1, B_1, C_1, D_1,

be the structural matrices of M_1. Then, there exists a transducer $M_2 \in [M]_{\sim_n}$ such that the structural matrices of M_2 are

$$A_2 = \left[\begin{array}{c:c} A_1 & 0_{n_1\times n'} \\ \hdashline 0_{n'\times n_1} & 0_{n'\times n'} \end{array}\right],\ B_2 = \left[\begin{array}{c} B_1 \\ \hdashline 0_{n'\times l} \end{array}\right],\ C_2 = \left[\begin{array}{c:c} C_1 & 0_{m\times n'} \end{array}\right],\ \text{and } D_2 = D_1,$$

where $n' = n - n_1$. The transducer M_2 constructed in this way is called the *trivial expansion* of M_1 to $\mathcal{L}_n$.

Proving that $M_2 \sim M$ is quite simple. We already know that there exists M_1 in the above conditions. Take $u = n_1+n$. Since $C_2A_2^i = [C_1A_1^i \quad 0_{m\times n'}]$, for $i = 0, 1, \ldots, u-1$, *i.e.*, $\Delta_{M_2}^{(u)} = [\Delta_{M_1}^{(u)} \quad 0_{um\times n'}]$, then, by Theorem 3.55, $M_2 \sim M_1$, which is equivalent to $M_2 \sim M$.

Next we count the number of transducers M_2' in $[M]_{\sim_n}$ that have $\hat{\Delta}_{M_2}$ as augmented diagnostic matrix. Basically, we study the possible choices for the structural matrices A_2', B_2', C_2' and D_2', of M_2', that satisfy the condition 2 of Theorem 3.55, and $\hat{\Delta}_{M_2} = \hat{\Delta}_{M_2'}$ (which implies condition 1). The choice for D_2' is obvious and unique from condition 2, as well as the choice for C_2' (from condition $\hat{\Delta}_{M_2} = \hat{\Delta}_{M_2'}$). It remains to compute how many choices does one have for A_2' such that the condition $\hat{\Delta}_{M_2} = \hat{\Delta}_{M_2'}$ is satisfied, and how many choices for B_2' such that $\hat{\Delta}_{M_2} = \hat{\Delta}_{M_2'}$ and condition 2 holds, *i.e.*, such that $\hat{\Delta}_M B_2 = \hat{\Delta}_{M_2'} B$. The following result gives the number of possible choices for A_2', and the proof gives the form of these matrices.

Proposition 4.6. *Let M_2 be an LFT with structural parameters $\ell, m, n \in \mathbb{N}$, and defined, as above, by structural matrices of the form*

$$A_2 = \left[\begin{array}{c:c} A_1 & 0_{n_2\times n'} \\ \hdashline 0_{n'\times n_2} & 0_{n'\times n'} \end{array}\right],\ B_2 = \left[\begin{array}{c} B_1 \\ \hdashline 0_{n'\times \ell} \end{array}\right],\ C_2 = \left[\begin{array}{c:c} C_1 & 0_{m\times n'} \end{array}\right],\ \textit{and } D_2 = D_1,$$

where $n_2 = \operatorname{rank}(\Delta_{M_2})$, $n' = n - n_2$, and the matrices A_1, B_1, C_1, and D_1 define a minimal LFT, M_1, equivalent to M_2. Then, the exact number of matrices $A \in \mathcal{M}_n(\mathbb{F}_q)$ such that $C_2A_2^i = C_2A^i$, for $i = 0, 1, \ldots, 2n-1$, is

$$q^{n\left(n-\operatorname{rank}(\Delta_{M_2})\right)}.$$

Proof. Let $A \in \mathcal{M}_n(\mathbb{F}_q)$ be such that $C_2A_2^i = C_2A^i$, for $i = 0, 1, \ldots, 2n-1$. And let $n' = n - n_2$, $E_1 \in \mathcal{M}_{n_2\times n_2}(\mathbb{F}_q)$, $E_2 \in \mathcal{M}_{n_2\times n'}(\mathbb{F}_q)$, $E_3 \in \mathcal{M}_{n'\times n_2}(\mathbb{F}_q)$, and $E_4 \in \mathcal{M}_{n'\times n'}(\mathbb{F}_q)$ be such that

$$A = \left[\begin{array}{c:c} E_1 & E_2 \\ \hdashline E_3 & E_4 \end{array}\right].$$

Then, from $C_2A_2^i = C_2A^{i-1}A = C_2A_2^{i-1}A$, for $i \in \{1, \ldots, 2n-1\}$ one gets that

$$\left[\begin{array}{cc} C_1A_1^i & 0_{m\times n'} \end{array}\right] = \left[\begin{array}{cc} C_1A_1^{i-1}E_1 & C_1A_1^{i-1}E_2 \end{array}\right], \text{ for } i \in \{1, \ldots, 2n-1\},$$

i.e.,

$$C_1A_1^i = C_1A_1^{i-1}E_1 \text{ and } C_1A_1^{i-1}E_2 = 0, \text{ for } i \in \{1, \ldots, 2n-1\}.$$

This is equivalent to

$$\Delta_{M_1}^{(2n-1)}A_1 = \Delta_{M_1}^{(2n-1)}E_1 \text{ and } \Delta_{M_1}^{(2n-1)}E_2 = 0,$$

or

$$\Delta_{M_1}^{(2n-1)}(A_1 - E_1) = 0 \text{ and } \Delta_{M_1}^{(2n-1)}E_2 = 0.$$

Since M_1 is minimal, by Lemma 3.54 and Corollary 3.53, $\operatorname{rank}(\Delta_{M_1}^{(2n-1)}) = \operatorname{rank}(\Delta_{M_1}) = n_2$ = number of columns of $\Delta_{M_1}^{(2n-1)}$. Therefore, $E_1 = A_1$ and $E_2 = 0$. Consequently, any matrix A with the same first n_2 rows as A_2 satisfies $C_2A_2^i = C_2A^i$, for $i = 0, 1, \ldots, 2n-2$, and those matrices A are the only ones that satisfy condition 2. Because the last $n - n_2$ rows of A can be arbitrarily chosen, and A has n columns, one gets that there are $q^{n(n-n_2)}$ matrices A that satisfy the required conditions. Since $n_2 = \operatorname{rank}(\Delta_{M_1}) = \operatorname{rank}(\Delta_{M_2})$ (because M_1 is minimal and equivalent to M_2), the result follows. □

As a consequence, going back to the question raised on the previous page, the number of possible choices for A_2' is $q^{n\left(n-\operatorname{rank}(\Delta_{M_2})\right)}$. Now, for each matrix A_2' such that $\hat{\Delta}_{M_2} = \hat{\Delta}_{M_2'}$, it remains to count the number of matrices B_2' that satisfy $\hat{\Delta}_{M_2}B_2 = \hat{\Delta}_{M_2}B_2'$.

Proposition 4.7. *Let M_2 be an LFT with structural parameters $\ell, m, n \in \mathbb{N}$, and defined by structural matrices of the form*

$$A_2 = \left[\begin{array}{c:c} A_1 & 0_{n_2\times n'} \\ \hdashline 0_{n'\times n_2} & 0_{n'\times n'} \end{array}\right],\ B_2 = \left[\begin{array}{c} B_1 \\ \hdashline 0_{n'\times \ell} \end{array}\right],\ C_2 = \left[\begin{array}{c:c} C_1 & 0_{m\times n'} \end{array}\right],\ \text{and } D_2 = D_1,$$

where $n_2 = \operatorname{rank}(\Delta_{M_2})$, $n' = n - n_2$, and the matrices A_1, B_1, C_1, and D_1 define a minimal LFT, M_1, equivalent to M_2. Given a matrix $A \in \mathcal{M}_n(\mathbb{F}_q)$ such that $C_2A_2^i = C_2A^i$, for $i = 0, 1, \ldots, 2n-1$, then the exact number of matrices $B \in \mathcal{M}_{n\times\ell}(\mathbb{F}_q)$ such that $C_2A^iB_2 = C_2A^iB$, for $i = 0, 1, \ldots, 2n-1$, is

$$q^{\ell\left(n-\operatorname{rank}(\Delta_{M_2})\right)}.$$

Proof. Let A be a matrix such that $C_2A_2^i = C_2A^i$, for $i = 0, 1, \ldots, 2n-1$, and B such that $C_2A^iB_2 = C_2A^iB$ for $i = 0, 1, \ldots, 2n-1$, *i.e.*, $\hat{\Delta}_{M_2}B_2 = \hat{\Delta}_{M_2}B$. Then, $\hat{\Delta}_{M_2}(B_2 - B) = 0$. Let B' be the submatrix formed by the first n_2 rows of $B_2 - B$. Since $\hat{\Delta}_{M_2} = \left[\begin{array}{cc} \Delta_{M_1}^{(2n)} & 0_{mn_2\times n'} \end{array}\right]$, it follows that $\Delta_{M_1}^{(2n)}B' = 0$. One knows that the columns of $\Delta_{M_1}^{(2n)}$ are linearly independent (because M_1 is minimal), then $\Delta_{M_1}^{(2n)}B' = 0$ implies $B' = 0$. Consequently, one can conclude that, to have a solution of $\hat{\Delta}_{M_2}(B_2 - B) = 0$, the first n_2 rows of B have to be equal to the first n_2 rows of B_2, and the last $n - n_2$ rows of B can be arbitrarily chosen. Since B has ℓ columns, that means that there are $q^{\ell(n-n_2)}$ matrices B in the required conditions. □

The number of possible choices for B_2' in the conditions above is $q^{\ell\left(n-\operatorname{rank}(\Delta_{M_2})\right)}$, thus the number of transducers in $[M]_{\sim_n}$ that have $\hat{\Delta}_{M_2}$ as augmented diagnostic matrix is

$$\left|E_{\hat{\Delta}_{M_2}}\right| = q^{(n+\ell)(n-r)},$$

where $r = \text{rank}(\Delta_{M_2})$.

From the results proven so far in this section, and since diagnostic matrices of LFTs in the same equivalence class have the same rank, the next theorem follows.

Theorem 4.8. *Let M be an LFT with structural parameters ℓ,m,$n \in \mathbb{N}$. Then*

$$|[M]_{\sim_n}| = \prod_{i=0}^{r-1} \left(q^n - q^i\right) q^{(n+\ell)(n-r)},$$

where $r = \text{rank}\,(\Delta_M)$.

Besides proving the previous theorem, the discussion presented gives a procedure to enumerate the LFTs in $[M]_{\sim_n}$, where M is an LFT of size $n \in \mathbb{N}$, namely:

1. Find a minimal transducer, $M_1 = (A_1, B_1, C_1, D_1)$, equivalent to M using, for example, the procedure presented in Section 3.4.

2. Construct the trivial expansion, $M_2 = (A_2, B_2, C_2, D_2)$, of M_1 to $\mathcal{L}_n$, and take $n_2 = \text{rank}(\Delta_{M_2})$.

3. Construct the set $\hat{S}_2$ of LFTs in $[M]_{\sim_n}$ that have $\hat{\Delta}_{M_2}$ as augmented diagnostic matrix, which, from previous discussion and proofs of Propositions 4.6 and 4.7, is given by

$$\hat{S}_2 = \left\{ \left(\left[\begin{array}{c:c} A_1 & 0 \\ \hdashline E_1 & E_2 \end{array}\right], \left[\begin{array}{c} B_1 \\ \hdashline F_1 \end{array}\right], C_2, D_2 \right) : E_1 \in \mathcal{M}_{n' \times n_2}, E_2 \in \mathcal{M}_{n' \times n'}, F_1 \in \mathcal{M}_{n' \times \ell} \right\}.$$

4. For each matrix $X \in GL_n(\mathbb{F}_q)$, determine the set $\hat{S}_X$ of transducers in $[M]_{\sim_n}$ that have $\hat{\Delta}_{M_2} X$ as augmented diagnostic matrix. From Proposition 4.1, that set is given by

$$\hat{S}_X = \left\{ \left(X^{-1}AX, X^{-1}B, CX, D\right) : (A, B, C, D) \in \hat{S}_2 \right\}.$$

The equivalence class of M in $\mathcal{L}_n$ is then given by

$$[M]_{\sim_n} = \bigcup_{X \in GL_n(\mathbb{F}_q)} \hat{S}_X.$$

Example 4.9. *Let $M = \langle \mathbb{F}_2, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta, \lambda \rangle$ be the LFT over $\mathbb{F}_2$ defined by the following structural matrices*

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Notice that

$$\Delta_M = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{\mathrm{T}},$$

and $\mathrm{rank}(\Delta_M) = 1 < 2 = \mathrm{size}(M)$. *Therefore, M is not minimal. Then, we can follow the steps presented above to enumerate the equivalence classe $[M]_{\sim_2}$. If M were minimal, we would jump directly to Step 4 (by letting $M_2 = M$ and $\hat{S}_2 = \{M\}$).*

1. *We construct a minimal LFT equivalent to M using the process described at the end of Section 3.4. Let $K = \begin{bmatrix} 0 & 1 \end{bmatrix}$ be a submatrix of Δ_M formed by* $\mathrm{rank}(\Delta_M) = 1$ *(linearly independent) row of Δ_M. A right inverse of K is $R = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathrm{T}}$. Therefore, the LFT $M_1 = \langle \mathbb{F}_2, \mathbb{F}_2^2, \mathbb{F}_2, \delta_1, \lambda_1 \rangle$ over $\mathbb{F}_2$ defined by the following structural matrices*

$$A_1 = KAR = [1], \qquad B_1 = KB = \begin{bmatrix} 1 \end{bmatrix},$$
$$C_1 = CR = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad D_1 = D = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

   is minimal and equivalent to M.

2. *The trivial expansion of M_1 to $\mathcal{L}_2$ is the transducer $M_2 = \langle \mathbb{F}_2, \mathbb{F}_2^2, \mathbb{F}_2^2, \delta_2, \lambda_2 \rangle$*

defined by the structural matrices:

$$A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ C_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},\ \text{and } D_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

3. *The set $\hat{S}_2$ of LFTs in $[M]_{\sim_2}$ that have $\hat{\Delta}_{M_2}$ as augmented diagnostic matrix is given on the left of Table 4.1.*

4. *In this step, we choose to enumerate just one of the sets $\hat{S}_X$, for $X \in GL_2(\mathbb{F}_q)$, because the others are obtained in a similar fashion. The set $\hat{S}_X$ of LFTs in $[M]_{\sim_2}$ that have $\hat{\Delta}_{M_2}X$ as augmented diagnostic matrix, for $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \in GL_2(\mathbb{F}_q)$, is given on the right of Table 4.1.*

4.3 Number of Equivalence Classes

Now that we already know how to evaluate the size of equivalence classes, it remains to show how to compute the number of equivalence classes in the set of non-trivial LFTs with structural parameters $\ell, m, n \in \mathbb{N}$. Let κ denote that number and $\mathsf{C}_{\ell,m,n}$ denote the number of canonical LFTs over $\mathbb{F}_q$ with structural parameters ℓ, m, n. From Section 4.1, one knows that each non-trivial equivalence class has exactly one canonical LFT. Consequently,

$$\kappa = \sum_{i=1}^{n} \mathsf{C}_{\ell,m,i}\,. \tag{4.2}$$

In the remaining of this section we deduce a recurrence relation that, given $\ell, m, n \in \mathbb{N}$, counts $\mathsf{C}_{\ell,m,n}$, and, therefore, using (4.2), allows to compute κ.

Let $\ell, m, n \in \mathbb{N}$, and consider the following notation:

- $\mathsf{L}_{\ell,m,n}$ denotes the total number of LFTs over $\mathbb{F}_q$ in $\mathcal{L}_{\ell,m,n}$;
- $\mathsf{T}_{\ell,m,n}$ denotes the number of trivial LFTs over $\mathbb{F}_q$ in $\mathcal{L}_{\ell,m,n}$;

$\hat{S}_2$			
A	B	C	D
$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\hat{S}_X$			
$X^{-1}AX$	$X^{-1}B$	CX	D
$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Table 4.1 – Enumeration of transducers in $\hat{S}_2$ and $\hat{S}_X$ of Example 4.9.

- $\mathsf{mL}_{\ell,m,n}$ denotes the number of non-trivial LFTs over $\mathbb{F}_q$ in $\mathcal{L}_{\ell,m,n}$ that are minimal;

- $\overline{\mathsf{m}}\mathsf{L}_{\ell,m,n}$ denotes the number of non-trivial LFTs over $\mathbb{F}_q$ in $\mathcal{L}_{\ell,m,n}$ that are not minimal.

It is obvious that

$$\mathsf{L}_{\ell,m,n} = q^{m\ell+n(\ell+m+n)} = \mathsf{T}_{\ell,m,n} + \overline{\mathsf{m}}\mathsf{L}_{\ell,m,n} + \mathsf{mL}_{\ell,m,n}\,.$$

The number of trivial transducers is easy to find: since an LFT is trivial when $C = 0$, the entries of the other matrices (A, B, and D) can take any value. Thus

$$\mathsf{T}_{\ell,m,n} = q^{n^2+\ell(m+n)}.$$

The set of non-trivial LFTs in $\mathcal{L}_{\ell,m,n}$ that are minimal is formed by the equivalence classes that have a canonical LFT. By Theorem 4.8, all such classes have the same cardinality. Let EC_n be the size of the equivalence class $[M]_{\sim_n}$, where M is a canonical transducer in $\mathcal{L}_{\ell,m,n}$. Then, also from Theorem 4.8, $\mathsf{EC}_n = \prod_{i=0}^{n-1}(q^n - q^i)$. Therefore,

$$\mathsf{mL}_{\ell,m,n} = \mathsf{EC}_n \cdot \mathsf{C}_{\ell,m,n} = \prod_{i=0}^{n-1}(q^n - q^i) \cdot \mathsf{C}_{\ell,m,n}\,.$$

Now, let us see how to determine $\overline{\mathsf{m}}\mathsf{L}_{\ell,m,n}$ for all $\ell, m, n \in \mathbb{N}$.

For $n = 1$, all the non-trivial LFTs are canonical. Therefore $\overline{\mathsf{m}}\mathsf{L}_{\ell,m,1} = 0$, and

$$\mathsf{C}_{\ell,m,1} = \mathsf{L}_{\ell,m,1} - \mathsf{T}_{\ell,m,1} = (q^m - 1)q^{\ell(m+1)+1}. \tag{4.3}$$

For $n = 2$, $\overline{\mathsf{m}}\mathsf{L}_{\ell,m,n}$ is the number of transducers in $\mathcal{L}_{\ell,m,2}$ that are equivalent to transducers in $\mathcal{L}_{\ell,m,1}$. Since, given a linear transducer M, $\operatorname{rank}(\Delta_M) = \operatorname{rank}(\Delta_{M'})$, where M' is a minimal LFT equivalent to M, using Theorem 4.8 we have a way to compute the number of LFTs in $\mathcal{L}_{\ell,m,n_2}$ that are equivalent to minimal transducers in

$\mathcal{L}_{\ell,m,n_1}$, for $n_2 \geq n_1$. Let $\mathsf{NM}_{\ell,n_1,n_2}$ be that value, that is,

$$\mathsf{NM}_{\ell,n_1,n_2} = \prod_{i=0}^{n_1-1} (q^{n_2} - q^i) \cdot q^{(n_2+\ell)(n_2-n_1)}.$$

Then,

$$\overline{\mathsf{m}}\mathsf{L}_{\ell,m,2} = \mathsf{C}_{\ell,m,1} \cdot \mathsf{NM}_{\ell,1,2} = \mathsf{C}_{\ell,m,1} \cdot (q^2 - 1) \cdot q^{\ell+2}.$$

For $n = 3$, the set of non-minimal LFTs if formed by the LFTs that are equivalent to minimal transducers in $\mathcal{L}_{\ell,m,1}$, and then ones that are equivalent to minimal transducers in $\mathcal{L}_{\ell,m,2}$. Therefore,

$$\begin{aligned} \overline{\mathsf{m}}\mathsf{L}_{\ell,m,3} &= \mathsf{C}_{\ell,m,1} \cdot \mathsf{NM}_{\ell,1,3} + \mathsf{C}_{\ell,m,2} \cdot \mathsf{NM}_{\ell,2,3} \\ &= \sum_{i=1}^{2} \mathsf{C}_{\ell,m,i} \cdot \mathsf{NM}_{\ell,i,3} = \sum_{i=1}^{2} \mathsf{C}_{\ell,m,i} \cdot \prod_{j=0}^{i-1} (q^3 - q^j) \cdot q^{(\ell+3)(3-i)}. \end{aligned}$$

This process can be generalised to get:

$$\overline{\mathsf{m}}\mathsf{L}_{\ell,m,n} = \sum_{i=1}^{n-1} \mathsf{C}_{\ell,m,i} \cdot \mathsf{NM}_{\ell,i,n}\,.$$

Therefore, given $\ell, m, n \in \mathbb{N}$, the number of canonical LFTs with structural parameters ℓ, m, n satisfies the following recurrence relation:

$$\begin{cases} \mathsf{C}_{\ell,m,1} &= (q^m - 1) q^{\ell(m+1)+1}, \\ \mathsf{C}_{\ell,m,n} &= \frac{1}{\mathsf{EC}_n} \cdot (\mathsf{L}_{\ell,m,n} - \mathsf{T}_{\ell,m,n} - \overline{\mathsf{m}}\mathsf{L}_{\ell,m,n})\,, \text{ for } n \geq 2, \end{cases}$$

and one has

- $\mathsf{L}_{\ell,m,n} = q^{m\ell+n(\ell+m+n)}$,
- $\mathsf{EC}_n = \prod_{i=0}^{n-1} (q^n - q^i)$,
- $\mathsf{T}_{\ell,m,n} = q^{n^2+\ell(m+n)}$,
- $\overline{\mathsf{m}}\mathsf{L}_{\ell,m,n} = \sum_{i=1}^{n-1} \mathsf{C}_{\ell,m,i} \cdot \mathsf{NM}_{\ell,i,n}$,

- $\mathsf{NM}_{\ell,i,n} = \prod_{j=0}^{i-1}(q^n - q^j) \cdot q^{(n+\ell)(n-i)}$,

and the number of non-trivial equivalence classes is given by

$$\kappa = \sum_{i=1}^{n} \mathsf{C}_{\ell,m,i},$$

as explained in the beginning of this section.

Example 4.10. *Using the recurrence relation above one gets, for example,*

$$\begin{aligned} |\mathcal{L}_{2,2,5}/\!\sim| &= \mathsf{C}_{2,2,1} + \mathsf{C}_{2,2,2} + \mathsf{C}_{2,2,3} + \mathsf{C}_{2,2,4} + \mathsf{C}_{2,2,5} \\ &= 384 + 7168 + 122880 + 2031616 + 33030144 \\ &= 35\,192\,192, \end{aligned}$$

$$\begin{aligned} |\mathcal{L}_{2,5,2}/\!\sim| &= \mathsf{C}_{2,5,1} + \mathsf{C}_{2,5,2} \\ &= 253\,952 + 42\,663\,936 \\ &= 42\,917\,888, \end{aligned}$$

and

$$\begin{aligned} |\mathcal{L}_{5,2,2}/\!\sim| &= \mathsf{C}_{5,2,1} + \mathsf{C}_{5,2,2} \\ &= 196\,608 + 29\,360\,128 \\ &= 29\,556\,736. \end{aligned}$$

Chapter 5

Equivalence Classes of Injective LFTs

In what follows we always assume $\tau \in \mathbb{N}_0$, unless otherwise stated.

5.1 Injectivity of LFTs

Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be an LFT over a field $\mathbb{F}$ with structural matrices A, B, C, D, and structural parameters $\ell, m, n \in \mathbb{N}$. Recall, from Chapter 3, that starting at a state s_0 and reading an input sequence $x_0x_1x_2\cdots$, one gets a sequence of states $s_0s_1s_2\cdots$ and a sequence of outputs $y_0y_1y_2\cdots$ satisfying the relations

$$s_{t+1} = \delta(s_t, x_t) = As_t + Bx_t, \tag{5.1}$$

$$y_t = \lambda(s_t, x_t) = Cs_t + Dx_t, \tag{5.2}$$

for all $t \geq 0$. Let

$$X(z) = \sum_{t\geq 0} x_t z^t, \quad Y(z) = \sum_{t\geq 0} y_t z^t, \quad S(z) = \sum_{t\geq 0} s_t z^t,$$

regarded as elements of the $\mathbb{F}[[z]]$-modules $\mathbb{F}[[z]]^\ell$, $\mathbb{F}[[z]]^m$, $\mathbb{F}[[z]]^n$, respectively, where $\mathbb{F}[[z]]$ is the ring of formal power series over $\mathbb{F}$. Multiplying equality (5.1) by z^t, and

adding the corresponding sides for all $t \geq 0$, one obtains:

$$\begin{aligned} \sum_{i\geq 0} s_{i+1}z^i = AS(z) + BX(z) &\Leftrightarrow (S(z) - s_0)z^{-1} = AS(z) + BX(z) \\ &\Leftrightarrow (I - Az)S(z) = s_0 + BzX(z). \end{aligned}$$

Since $(I - Az) \in \mathcal{M}_n(\mathbb{F})[z]$ is invertible in $\mathcal{M}_n(\mathbb{F})[[z]]$, one can rewrite the above equality as follows:

$$S(z) = (I - Az)^{-1}s_0 + (I - Az)^{-1}BzX(z). \tag{5.3}$$

Analogously, multiplying equality (5.2) by z^t, and adding for all $t \geq 0$, one gets:

$$Y(z) = CS(z) + DX(z).$$

Therefore, using (5.3),

$$Y(z) = G(z)s_0 + H(z)X(z), \tag{5.4}$$

where

$$G(z) = C(I - Az)^{-1} \quad \text{and} \quad H(z) = C(I - Az)^{-1}Bz + D. \tag{5.5}$$

Notice that, since $(I - Az)$ is invertible in $\mathcal{M}_n(\mathbb{F})[[z]]$ and $(I - Az)^{-1} = \sum_{n\geq 0} A^n z^n$, from (5.3), one gets:

$$S(z) = \sum_{n\geq 0} A^n s_0 z^n + \sum_{n\geq 0} A^n BX(z)z^{n+1},$$

which gives equality (3.4) of Theorem 3.43. Analogously, from (5.4) one gets:

$$Y(z) = C\sum_{n\geq 0} A^n s_0 z^n + \left(C\sum_{n\geq 0} A^n Bz^{n+1} + D\right)X(z).$$

This proves the validity of (3.5), and, consequently, Theorem 3.43 is proven[1].

Tao [Tao09] calls the matrices $G \in \mathcal{M}_{m\times n}(\mathbb{F})[[z]]$ and $H \in \mathcal{M}_{m\times \ell}(\mathbb{F})[[z]]$, respectively, *free response matrix* and *transfer function matrix* of the transducer. This choice of terminology (adopted below) is due to Massey and Slain [MS68]. The following result was presented by Zongduo and Dingfeng [ZD96] without proof.

Theorem 5.1. *Let $M = \langle \mathbb{F}^\ell, \mathbb{F}^m, \mathbb{F}^n, \delta, \lambda \rangle$ be a linear finite transducer with structural matrices A, B, C and D. Let $H(z)$ be its transfer function matrix. Then, $H(z)$ is of the form*

$$\frac{1}{f(z)} \sum_{i=0}^{n} H_i z^i,$$

where $H_i \in \mathcal{M}_{m\times\ell}(\mathbb{F})$, and $f(z) \in \mathbb{F}[z]$ is such that $f(0) = 1$.

Proof. Since

$$(I - Az)^{-1} = \frac{(I - Az)^*}{|I - Az|},$$

where $P^* = \mathrm{adj}(P)$, one gets, from (5.5), that

$$H(z) = C\frac{(I - Az)^*}{|I - Az|}Bz + D = \frac{1}{|I - Az|}\left(C(I - Az)^*Bz + |I - Az|D\right).$$

Let $f(z) = |I - Az|$. Thus $f(0) = 1$, because the independent term of $|I - Az|$ is 1. Since the entries of the matrix $I - Az$ are polynomials of degree ≤ 1 and $A \in \mathcal{M}_n(\mathbb{F})$, the entries of the matrix $(I - Az)^*$ are polynomials of degree $\leq n - 1$. Also, the degree of the polynomial $|I - Az|$ is $\leq n$. Therefore, the entries of the matrix $C(I - Az)^*Bz + |I - Az|D$ are polynomials of degree $\leq n$. Since a matrix of polynomials can be interpreted as a polynomial whose coefficients are matrices, the result follows. □

From the proof of the last theorem, one knows that

$$H(z) = \frac{1}{f(z)}\left(C(I - Az)^*Bz + f(z)D\right), \tag{5.6}$$

[1]In some contexts, $X(z)$, $Y(z)$ and $S(z)$ as defined above may be known as the $z-$ transformation of the sequences $x_0x_1x_2\cdots$, $y_0y_1y_2\cdots$ and $s_0s_1s_2\cdots$, respectively.

where $f(z) = |I - Az|$. Consider the multiplicatively closed set

$$\mathcal{S} = \{1 + zb(z) \mid b(z) \in \mathbb{F}[z]\},$$

and let $\mathbb{F}[z]_{\mathcal{S}}$ be the localisation of $\mathbb{F}[z]$ relative to S, *i.e.*,

$$\mathbb{F}[z]_{\mathcal{S}} = \left\{ \frac{f}{s} \;\middle|\; f \in \mathbb{F}[z], s \in \mathcal{S} \right\}.$$

Then, the previous result states that the transfer function matrix of an LFT is in $\mathcal{M}(\mathbb{F}[z]_{\mathcal{S}})$. It is known that $\mathbb{F}[z]_{\mathcal{S}}$ is a principal ideal domain, and z is its unique irreducible element, up to units [AM69]. Then, from Theorem 2.21, it follows that every matrix $H(z) \in \mathcal{M}(\mathbb{F}[z]_{\mathcal{S}})$ with rank r is equivalent to a "diagonal" matrix of the form

$$\mathcal{D}_{n_0,n_1,\ldots,n_u} = diag(I_{n_0}, zI_{n_1}, \ldots, z^u I_{n_u}, 0, \ldots, 0),$$

where $n_i \geq 0$, for $0 \leq i \leq u$, $n_u \neq 0$ unless $H(z) = 0$, and $\sum_{i=0}^{u} n_i = r$. In order to facilitate the statement of the next result, we put $n_i = 0, \forall i > u$. The Smith normal form of $H(z)$ is used, in the next theorem, to give two necessary and sufficient conditions for an LFT to be injective with some delay $\tau \in \mathbb{N}_0$. This result is a restatement of the results about τ-injectivity presented by Zongduo and Dingfeng in [ZD96, Theorem 1 and Theorem 2].

Theorem 5.2. *Let $\mathcal{X}, \mathcal{Y}$ and S be vector spaces over a field $\mathbb{F}$, with dimensions ℓ, m, $n \in \mathbb{N}$, respectively. Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be an LFT, and let $H \in \mathcal{M}_{m\times\ell}(\mathbb{F}[z]_{\mathcal{S}})$ be its transfer function matrix. Let $\mathcal{D} = \mathcal{D}_{n_0,n_1,\ldots,n_u}$ be the Smith normal form of H, and assume that $n_u \neq 0$. Then, the following conditions are equivalent:*

i. M is injective with delay τ;

ii. $\sum_{i=0}^{\tau} n_i = \ell$;

iii. there is $H' \in \mathcal{M}_{\ell\times m}(\mathbb{F}[z]_{\mathcal{S}})$ such that $H'H = z^\tau I$.

Moreover, if M is τ-injective, for some $\tau \in \mathbb{N}_0$, then it is u-injective.

Proof. ($i. \Rightarrow ii.$) Suppose that $\sum_{i=0}^{\tau} n_i \neq \ell$, *i.e.*, $\sum_{i=0}^{\tau} n_i < \ell$. Let

$$X = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^{\mathrm{T}} \in \mathcal{M}_{\ell\times 1}\left(\mathbb{F}[[z]]\right).$$

Then $\mathcal{D}X = \mathbf{0}_{m\times 1}$. If $P \in GL_m(\mathbb{F}[z]_{\mathcal{S}})$ and $N \in GL_\ell(\mathbb{F}[z]_{\mathcal{S}})$ are the matrices such that $\mathcal{D} = PHN$, then $HNX = \mathbf{0}_{m\times 1}$. Putting $X' = NX$, from (5.4) one gets that $\lambda(0, X') = HX' = \mathbf{0}_{m\times 1} = \lambda(0, \mathbf{0}_{\ell\times 1})$. Since $X' \neq \mathbf{0}_{\ell\times 1}$, it follows that M is not injective with delay τ.

($ii. \Rightarrow iii.$) The hypothesis implies that, in $\mathcal{D}$, one has $\tau \geq u$ and that there are no null columns. Take, again, P and N to be the invertible matrices such that $\mathcal{D} = PHN$, and let

$$\mathcal{D}' = diag(z^{\tau} I_{n_0}, z^{\tau-1} I_{n_1}, \ldots, z^{\tau-u} I_{n_u}) \in \mathcal{M}_{\ell\times m}(F[z]_{\mathcal{S}}).$$

Then $\mathcal{D}'\mathcal{D} = z^{\tau} I$, and consequently $\mathcal{D}'PHN = z^{\tau} I$. From this it follows that

$$\mathcal{D}'PH = z^{\tau} N^{-1} = N^{-1} z^{\tau} I.$$

Hence $(N\mathcal{D}'P)H = z^{\tau} I$.

($iii. \Rightarrow i.$) Let s be a state of M and X, X' two input sequences such that $\lambda(s, X) \equiv \lambda(s, X') \pmod{z^{\tau+1}}$. Assume that there is $H' \in \mathcal{M}_{\ell\times m}(F[z]_{\mathcal{S}})$ such that $H'H = z^{\tau} I$. Then,

$$\begin{aligned} \lambda(s, X) \equiv \lambda(s, X') \pmod{z^{\tau+1}} &\Leftrightarrow Gs + HX \equiv Gs + HX' \pmod{z^{\tau+1}} \\ &\Leftrightarrow HX \equiv HX' \pmod{z^{\tau+1}} \\ &\Leftrightarrow H(X - X') \equiv 0 \pmod{z^{\tau+1}}. \end{aligned}$$

This implies, from $H'H = z^{\tau} I$, that $z^{\tau} I(X - X') \equiv 0 \pmod{z^{\tau+1}}$. Consequently, $X \equiv X' \pmod{z}$, and, therefore, M is injective with delay τ. The last sentence in the statement of the theorem follows from $i. \Leftrightarrow ii.$, and the fact that $n_i = 0$, for all $i > u$. □

Corollary 5.3. *Let M be a linear finite transducer in the conditions of the previous theorem. Then, M is injective with some delay if and only if D has maximal rank, which, when $m = \ell$, is equivalent to* $\det(H) \neq 0$.

Remember, from Theorem 5.1, that $H(z) \in \mathcal{M}(\mathbb{F}[z]_{\mathcal{S}})$ is of the form

$$\frac{1}{f(z)} \sum_{i=0}^{n} H_i z^i,$$

where $H_i \in \mathcal{M}_{m\times\ell}(\mathbb{F})$, and $f(z) \in \mathbb{F}[z]$ is such that $f(0) = 1$. Since units are irrelevant in the Smith normal form computation, the invariant factors of $H(z)$ can be obtained from the invariant factors of the matrix $f(z)H(z) \in \mathcal{M}_{m\times\ell}(\mathbb{F})$ using the following result.

Proposition 5.4. *Let $\mathcal{D}_{fH} = diag(d'_1, d'_2, \ldots, d'_r, 0, \ldots, 0)$ be the* SNF *of $f(z)H(z)$ in $\mathcal{M}(\mathbb{F}[z])$ and $\mathcal{D}_H = diag(d_1, d_2, \ldots, d_r, 0, \ldots, 0)$ the* SNF *of $H(z)$ in $\mathcal{M}(\mathbb{F}[z]_{\mathcal{S}})$. Then,*

$$\forall i \in \{1, \ldots, r\},\ d_i = \gcd(d'_i, z^u), \tag{5.7}$$

where $r = \operatorname{rank}(H(z)) = \operatorname{rank}(f(z)H(z))$ and z^u is the biggest power of z that divides d'_r.

Proof. Let $\mathcal{D}_{fH} = diag(d'_1, d'_2, \ldots, d'_r, 0, \ldots, 0)$ be the SNF of $f(z)H(z)$ in $\mathcal{M}(\mathbb{F}[z])$. Then, the invariant factors d'_i, for $i \in \{1, \ldots, r\}$, are of the form $z^{m_i}\alpha$, where $m_i \geq 0$ and $\alpha \in \mathcal{S}$. Since α is a unit in $\mathbb{F}[z]_S$, the Smith normal form of $f(z)H(z)$ in $\mathcal{M}(\mathbb{F}[z]_{\mathcal{S}})$ is $diag(z^{m_1}, z^{m_2}, \ldots, z^{m_r}, 0, \ldots, 0)$. Furthermore, $f(z)$ is also a unit in $\mathbb{F}[z]_{\mathcal{S}}$. Consequently, the matrices $f(z)H(z)$ and $H(z)$ have the same Smith normal form in $\mathcal{M}(\mathbb{F}[z]_{\mathcal{S}})$. The result then follows. □

Using the previous result and condition ii. of Theorem 5.2, we have written a `Python` function, `IsInjective(A,B,C,D,tau)`, which tests if an LFT over $\mathbb{F}_2$, defined by its structural matrices, `A`, `B`, `C`, `D`, is `tau`-injective, for `tau` in $\mathbb{N}_0$. The source code of this function is presented in Listing 5.1.

```
1  def IsInjective(A, B, C, D, tau):
2      Ring = GF(Integer(2))['z']
3      (z,) = Ring._first_ngens(1)
4      poly = identity_matrix(A.nrows()) - A*z
5      fH = C*poly.adjoint()*B*z + poly.det()*D
6      D_fH = fH.elementary_divisors()
7      D_H = [i.gcd(z**(tau+1)) for i in D_fH if i != 0]
8      return B.ncols() == len([j for j in D_H if j <= z**tau])
```

Listing 5.1 – Testing the injectivity.

The algorithm starts by defining the ring $\mathbb{F}_2[z]$ (line 2), and z as a variable in that ring (line 3). The expression `identity_matrix(A.nrows())`, as the name suggests, returns the identity matrix whose size is the number of rows of A. The matrix $f(z)H(z)$ is then computed using the expression (5.6), and the algorithm uses functions `adjoint` and `det`, to compute the adjoint and the determinant of a matrix, respectively (line 5). The invariant factors of $f(z)H(z)$ are computed using the function `elementary_divisors` (line 6). Since, to check if condition ii. of Theorem 5.2 is verified one just needs to count the invariant factors of $H(z)$ that are less or equal to $z^{\texttt{tau}}$, we apply Proposition 5.4 in the algorithm, replacing z^u by $z^{\texttt{tau}+1}$ in expression (5.7) (line 7). The algorithm then returns `True` if the number of invariant factors of $H(z)$ which divide $z^{\texttt{tau}}$ is equal to ℓ, *i.e.*, is equal to the number of columns of the matrix B. It returns `False` otherwise.

The input parameters A, B, C, D are matrices created using the `Sage` function `matrix`. For example, the matrices

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},\ \text{and } D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix},$$

with entries in $\mathbb{F}_2$, can be constructed by:

```
> A = matrix(GF(2),[[0,1],[1,0]])

> B = matrix(GF(2),[[1,0],[0,1]])

> C = matrix(GF(2),[[1,1],[1,0],[0,1]])

> D = matrix(GF(2),[[0,0],[1,0],[0,0]]).
```

Let M be the LFT defined by the structural matrices A, B, C, D as above. Using function `IsInjective(A,B,C,D,tau)`, it is easy to check if M is 0-injective or 1-injective, for example.

```
> IsInjective(A,B,C,D,0)

False

> IsInjective(A,B,C,D,1)

True
```

5.2 Number of Injective Equivalence Classes

In this section we show how to estimate the number of τ-injective equivalence classes in $\mathcal{L}_{\ell,m,n}$, for $\tau \in \mathbb{N}_0$, and a triple of structural parameters $\ell, m, n \in \mathbb{N}$.

Let $\mathcal{I}_\tau$ be the subset of τ-injective equivalence classes in $\mathcal{L}_{\ell,m,n}/\!\sim$, *i.e.*,

$$\mathcal{I}_\tau = \{[M] \in \mathcal{L}_{\ell,m,n}/\!\sim \mid M \text{ is } \tau\text{-injective}\}\,.$$

Given $[M] \in \mathcal{L}_{\ell,m,n}/\!\sim$, let $p_{[M]}$ be the probability that an LFT in $\mathcal{L}_{\ell,m,n}$ is in class $[M]$, that is,

$$p_{[M]} = \frac{|[M]|}{|\mathcal{L}_{\ell,m,n}|}.$$

The following result gives us a way to get an approximate value for $|\mathcal{I}_\tau|$, using uniformly

random generated LFTs.

Proposition 5.5. *Let $\mathcal{R}$ be a multiset of uniformly random generated LFTs in $\mathcal{L}_{\ell,m,n}$, for a given triple of structural parameters $\ell, m, n \in \mathbb{N}$. Let $\tau \in \mathbb{N}_0$. Then*

$$|\mathcal{I}_\tau| \approx \frac{1}{|\mathcal{R}|} \sum_{M \in \mathcal{R}} \mu_{[M]},$$

where

$$\mu_{[M]} = \begin{cases} \frac{1}{p_{[M]}}, & \text{if } [M] \in \mathcal{I}_\tau, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. In what follows, let $\mathcal{E} = \mathcal{L}_{\ell,m,n}/\!\sim$. Trivially

$$|\mathcal{I}_\tau| = \sum_{[M] \in \mathcal{I}_\tau} 1 = \sum_{[M] \in \mathcal{I}_\tau} p_{[M]} \frac{1}{p_{[M]}} = \sum_{[M] \in \mathcal{E}} p_{[M]} \mu_{[M]}.$$

Let $\eta_{[M]}$ be the number of occurrences in $\mathcal{R}$ of transducers that belong to a class $[M] \in \mathcal{E}$. One knows that $p_{[M]} \approx \frac{\eta_{[M]}}{|\mathcal{R}|}$. Consequently,

$$|\mathcal{I}_\tau| \approx \sum_{[M] \in \mathcal{E}} \frac{\eta_{[M]}}{|\mathcal{R}|} \mu_{[M]} = \frac{1}{|\mathcal{R}|} \sum_{[M] \in \mathcal{E}} \eta_{[M]} \mu_{[M]} = \frac{1}{|\mathcal{R}|} \sum_{M \in \mathcal{R}} \mu_{[M]}.$$

□

From the previous result, computing an estimate of $|\mathcal{I}_\tau|$, from a sample of uniformly random generated LFTs, requires computing the size of each corresponding equivalence class, besides checking if the transducer is τ-injective. Recall that, from Theorem 4.8, given an LFT over $\mathbb{F}_q$, M, with structural parameters $\ell, m, n \in \mathbb{N}$, the size of its equivalence class is given by:

$$|[M]_{\sim_n}| = \prod_{i=0}^{r-1} \left(q^n - q^i\right) \cdot q^{(n+\ell)(n-r)}, \tag{5.8}$$

where $r = \operatorname{rank}(\Delta_M)$. Therefore, given an LFT, computing the size of its equivalence class in $\mathcal{L}/\!\sim_n$ is reduced to the construction of the associated diagnostic matrix and

the determination of its rank. Leveraging `Sage`'s ability to deal with matrices, we have written a `Python` function, `EquivClassSize(A,B,C,D)`, that computes the size of an equivalence class using expression (5.8) for $q = 2$. The input parameters of this function are the structural matrices `A,B,C,D` of an LFT in the chosen class. The source code of this function is in Listing 5.2.

```
1   def EquivClassSize(A, B, C, D):
2       l = B.ncols()
3       m = C.nrows()
4       n = A.nrows()
5       K = copy.deepcopy(C)
6       for j in {1,...,n-1}:
7           K = K.stack(K * A)
8       r = K.rank()
9       size = 1
10      for j in {0,...,r-1}:
11          size = size * (2**n - 2**j)
12      size = size * 2**((n + l) * (n - r))
13      return size
```

Listing 5.2 – Determining the size of equivalence classes.

The algorithm starts by determining the structural parameters ℓ, m, n that are computed using `Sage` functions `nrows` and `ncols` (lines 2–4). To compute the value of r in (5.8), it calls functions `stack` and `rank`. The first is used to create the LFT diagnostic matrix (lines 5–7), and the second is used to determine the rank of that matrix (line 8). The size of the equivalence class is then easily obtained through a loop (lines 9–12).

Example 5.6. *Let M be the LFT over $\mathbb{F}_2$ defined by the structural matrices*

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad \textit{and } D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Using the previous function one gets $|[M]_{\sim_3}| = 168$.

Since, from the previous section, we already have a function that checks if an LFT is τ-injective, we are now able to give a `Python` function, `EstCountInjective` (Listing 5.3), that estimates the number of `tau`-injective equivalence classes. The parameters of this function are

- `nr`: the sample size (size of $\mathcal{R}$ in the previous result),
- `l,m,n`: the structural parameters,
- `tau`: the delay,

and it calls the following three functions:

- `IsInjective(A,B,C,D,tau)`: the function defined in Section 5.1;
- `Probability(A,B,C,D)`: a function (Listing 5.4) that, given the structural matrices of an LFT, M, returns $p_{[M]}$ using the function `EquivClassSize`;
- `RandomLFT(l,m,n)`: a function that, given the structural parameters `l,m,n`, returns the structural matrices $A \in \mathcal{M}_n(\mathbb{F}_2)$, $B \in \mathcal{M}_{n\times\ell}(\mathbb{F}_2)$, $C \in \mathcal{M}_{m\times n}(\mathbb{F}_2)$, and $D \in \mathcal{M}_{m\times\ell}(\mathbb{F}_2)$ of a non-trivial LFT. The entries of those matrices are uniformly random generated using the `Python` module named `random`. The source code of this function is also presented in Listing 5.4.

```
1  def EstCountInjective(nr, l, m, n, tau):
2      count = 0
3      for i in {1, ..., nr}:
4          A, B, C, D = RandomLFT(l, m, n)
5          if IsInjective(A, B, C, D, tau):
6              count = count + 1/Probability(A, B, C, D)
7      return count/nr
```

Listing 5.3 – Estimating the number of non-equivalent LFTs.

Given an input, the algorithm in Listing 5.3 starts by initialising the variable `count` with the value 0. Then, at each iteration of the loop, it uniformly random generates an LFT, M, and, if M is injective with delay `tau`, it adds the value of $\mu_{[M]}$ to the variable `count` (lines 3–6). In this way, when the loop is finished, one has `count` $= \sum_{M \in \mathcal{R}} \mu_{[M]}$, where $\mathcal{R}$ is the set of the `nr` uniformly random generated LFTs. It returns `count/nr`, that is, an estimate for $|\mathcal{I}_{\texttt{tau}}|$.

```
1  def Probability(A, B, C, D):
2      l = B.ncols()
3      m = C.nrows()
4      n = A.nrows()
5      sizeLn = 2**(n*(n+l+m)+m*l) - 2**(n**2+l*(m+n))
6      return (1.0*EquivClassSize(A, B, C, D)) / sizeLn
7  def RandomLFT(l, m, n):
8      A = matrix(GF(2), n, [random.randint(0,1) for _ in range(n*n)])
9      B = matrix(GF(2), n, l, [random.randint(0,1) for _ in range(n*l)])
10     C = matrix(GF(2), m, n, [random.randint(0,1) for _ in range(m*n)])
11     D = matrix(GF(2), m, l, [random.randint(0,1) for _ in range(m*l)])
12     while C == matrix(GF(2), m, n):
13         C = matrix(GF(2), m, n, [random.randint(0,1) for _ in range(m*n)])
14     return A, B, C, D
```

Listing 5.4 – Auxiliary functions.

5.3 Percentage of Injective Equivalence Classes

In this section we estimate the probability of getting an injective equivalence class through uniform random generation of LFTs, *i.e.*, we want to estimate

$$\frac{|\mathcal{I}_\tau|}{|\mathcal{L}_{\ell,m,n}/\!\sim|}.$$

From the last section we already have a `Python` function, `EstCountInjective`, that computes an estimate of $|\mathcal{I}_\tau|$. Writing a `Python` function to compute $|\mathcal{L}_{\ell,m,n}/\!\sim|$ is not hard, if one uses the fact that each equivalence class has exactly one canonical LFT. Remember, from Section 4.3, that the number of canonical LFTs with structural parameters $\ell, m, n \in \mathbb{N}$, denoted $\mathsf{C}_{\ell,m,n}$, satisfies the following recurrence relation:

$$\begin{cases} \mathsf{C}_{\ell,m,1} &= (q^m - 1)q^{\ell(m+1)+1}, \\ \mathsf{C}_{\ell,m,n} &= \frac{1}{\mathsf{EC}_n} \cdot \left(\mathsf{L}_{\ell,m,n} - \mathsf{T}_{\ell,m,n} - \overline{\mathsf{m}}\mathsf{L}_{\ell,m,n}\right), \text{ for } n \geq 2, \end{cases}$$

where

- $\mathsf{L}_{\ell,m,n} = q^{m\ell+n(\ell+m+n)}$,
- $\mathsf{EC}_n = \prod_{i=0}^{n-1}(q^n - q^i)$,
- $\mathsf{T}_{\ell,m,n} = q^{n^2+\ell(m+n)}$,
- $\overline{\mathsf{m}}\mathsf{L}_{\ell,m,n} = \sum_{i=1}^{n-1} \mathsf{C}_{\ell,m,i} \cdot \mathsf{NM}_{\ell,i,n}$,
- $\mathsf{NM}_{\ell,i,n} = \prod_{j=0}^{i-1}(q^n - q^j) \cdot q^{(n+\ell)(n-i)}$.

Hence, we have a `Python` function, `CountCT(l,m,n)` (Listing 5.5), that, given a triple of structural parameters `l,m,n`, computes the number of canonical LFTs in $\mathcal{L}_{\mathtt{l},\mathtt{m},\mathtt{n}}$, using the recurrence relation above.

```
1  def CountCT(l, m, n):
2      if n = 1:
3          return (2**m - 1) * 2**(l * (m + 1) + 1)
4      else:
5          EC = 1
6          for i in {0, ..., n - 1}:
7              EC = EC * (2**n - 2**i)
8          LT = 2**(m * l + n * (l + m + n))
```

```
    TT = 2**(n**2 + l*(m+n))
    TNM = 0
    for i in {1,...,n-1}:
        NM = 2**(n+l)*(n-i)
        for j in {0,...,i-1}:
            NM = NM*(2**n - 2**j)
        TNM = TNM + CountCT(l,m,i)*NM
return (LT - TT - TNM)/EC
```

Listing 5.5 – Counting the number of canonical LFTs.

We also know, from the same section, that the total number of equivalence classes is given by

$$|\mathcal{L}_{\ell,m,n}/\!\sim| = \sum_{i=1}^{n} C_{\ell,m,i}.$$

Thus, using the functions `EstCountInjective` and `CountCT`, we can now define an elementary `Python` function that estimates the percentage of τ-injective equivalence classes, for $\tau \in \mathbb{N}_0$, and a set of structural parameters $\ell, m, n \in \mathbb{N}$. Listing 5.6 comprises the source code of such a function. Its parameters are the same as those of the function `EstCountInjective`.

```
def EstPercInjective(nr,l,m,n,tau):
    EC = 0
    for i in {1,...,n}:
        EC = EC + CountCT(l,m,i)
    return EstCountInjective(nr,l,m,n,tau)/EC
```

Listing 5.6 – Estimating the percentage of injective equivalence classes.

5.4 Experimental Results

In this section we present some experimental results on the number and percentage of τ-injective equivalent classes of LFTs over $\mathbb{F}_2$, for some values of $\tau \in \mathbb{N}_0$. Recall that if an LFT is τ-injective for some $\tau \in \mathbb{N}_0$, then it is ω-injective, and the converse is also true (Theorem 3.29).

For each triple of structural parameters ℓ, m, n, with $\ell \in \{1, \ldots, 5\}$, $m = 5$ and $n \in \{1, \ldots, 10\}$, we uniformly random generated a sample of 20 000 LFTs over $\mathbb{F}_2$. And, for each one of those samples, we estimated the number and percentage of τ-injective equivalence classes, for $\tau \in \{0, 1, \ldots, 10\}$, using the `Python` functions `EstCountInjective` and `EstPercInjective`, respectively. The size of each sample is sufficient to ensure the statistical significance with a 99% confidence level within a 1% error margin. The sample size is calculated with the formula $N = (\frac{z}{2\epsilon})^2$, where z is obtained from the normal distribution table such that $P(-z < Z < z) = \gamma$, ϵ is the error margin, and γ is the desired confidence level.

In Table 5.1, we present the obtained estimates of the number of 10-injective equivalence classes when $m = 5$, and n, ℓ range in $\{1, \ldots, 10\}$ and $\{1, \ldots, 5\}$, respectively. We chose to show the results for $\tau = 10$ because this value is large enough to draw conclusions about the number of ω-injective equivalence classes.

		ℓ				
		1	2	3	4	5
n	1	3.91×10^{03}	2.42×10^{05}	1.44×10^{07}	7.66×10^{08}	2.97×10^{10}
	2	3.34×10^{05}	4.17×10^{07}	5.13×10^{09}	5.92×10^{11}	5.29×10^{13}
	3	2.45×10^{07}	6.15×10^{09}	1.54×10^{12}	3.70×10^{14}	7.39×10^{16}
	4	1.66×10^{09}	8.45×10^{11}	4.26×10^{14}	2.10×10^{17}	9.24×10^{19}
	5	1.10×10^{11}	1.12×10^{14}	1.13×10^{17}	1.14×10^{20}	1.05×10^{23}
	6	7.17×10^{12}	1.45×10^{16}	2.96×10^{19}	5.97×10^{22}	1.15×10^{26}
	7	4.61×10^{14}	1.87×10^{18}	7.64×10^{21}	3.10×10^{25}	1.22×10^{29}
	8	2.96×10^{16}	2.40×10^{20}	1.96×10^{24}	1.60×10^{28}	1.28×10^{32}
	9	1.90×10^{18}	3.08×10^{22}	5.04×10^{26}	8.24×10^{30}	1.33×10^{35}
	10	1.22×10^{20}	3.95×10^{24}	1.29×10^{29}	4.23×10^{33}	1.37×10^{38}

Table 5.1 – Approximated values for the number of injective equivalence classes when $m = 5$ and $\tau = 10$.

From the results obtained, one can observe an exponential growth on the number of

10-injective equivalence classes, as n and ℓ increase. Consequently, the number of ω-injective equivalence classes also grows exponentially.

The approximate values obtained for the percentage of τ-injective equivalence classes, for $\ell \in \{2, 3, 4, 5\}$, are presented in Figures 5.1–5.3 (the tables of results can be seen in Appendix A). We have fitted a surface to these results[2]. The purpose of this fitting is merely to get a better 3D visualisation of the percentage variation. Figure 5.1 shows a 3D representation of the estimates obtained, and corresponding surface, for $\ell = 2$, from two different perspectives.

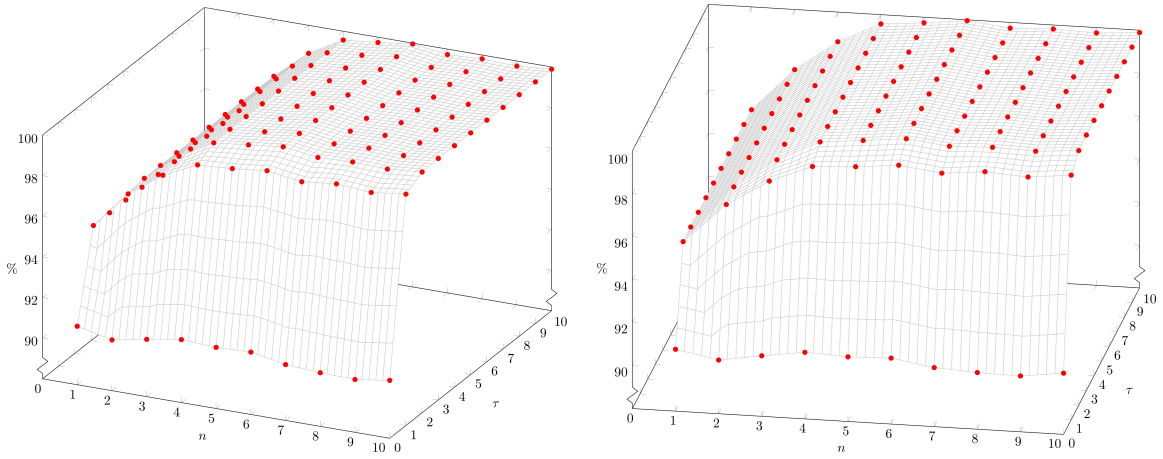


Figure 5.1 – Variation on the percentage of τ-injective equivalence classes for $\ell = 2$, $m = 5$, and several values of n and τ (from two different perspectives).

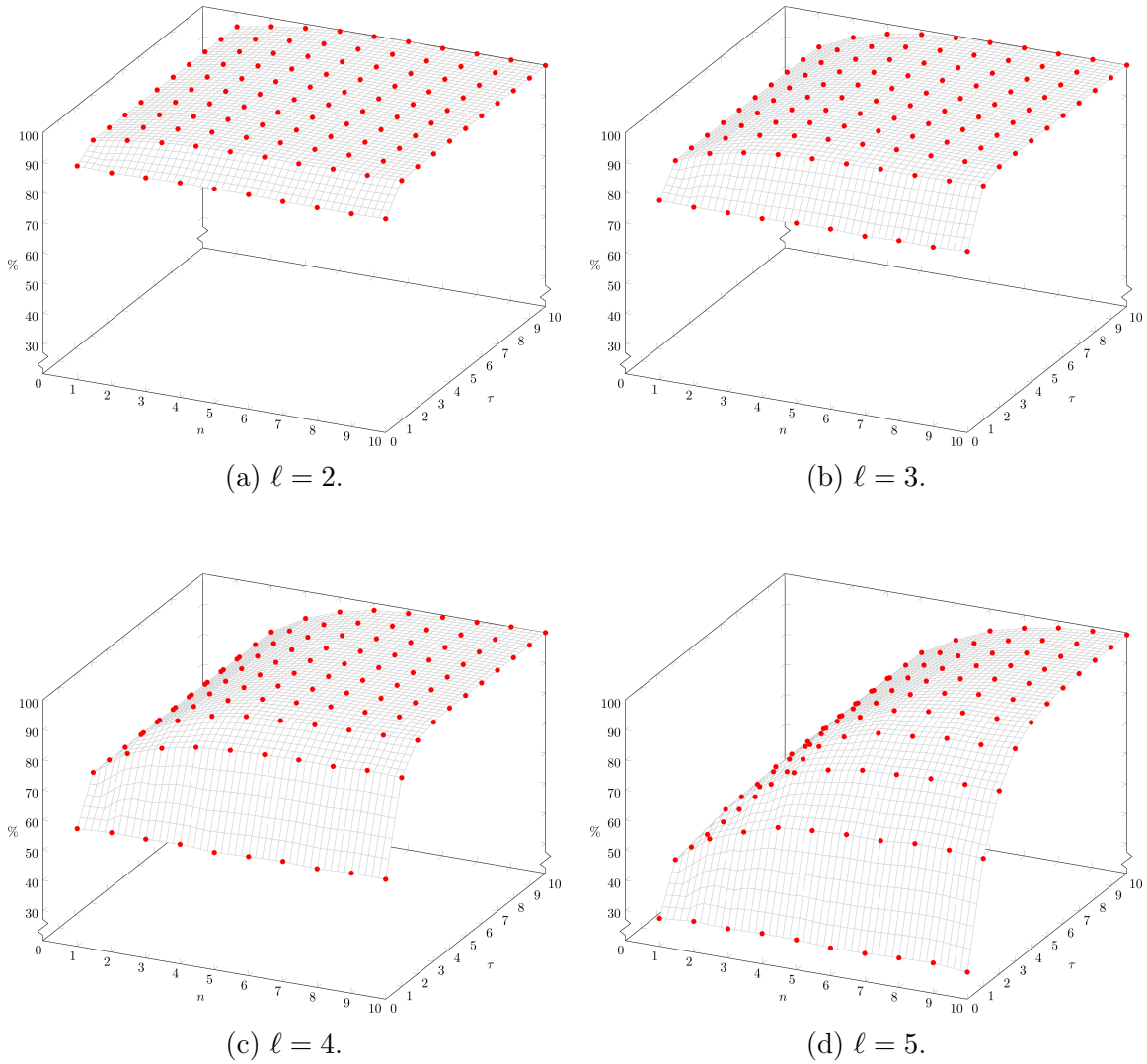
The results obtained allow some observations:

- when $n = 1$, the percentage of τ-injective equivalence classes is already above 90%, for $\tau \in \{0, 1, \ldots, 10\}$;
- when n increases, there is a significant increase in the percentage of τ-injective equivalence classes, for $\tau \geq 1$. It goes from values around 96% to values near 100%.

This suggests that, in this case, there is a very high probability of a uniformly random generated LFT be ω-injective.

[2]We used `Octave` function `griddata` and its triangulation-based linear interpolation method.

Figure 5.2 presents the results obtained for $\ell \in \{2, 3, 4, 5\}$. A different perspective of the same representations can be seen in Figure 5.3.



(a) $\ell = 2$. (b) $\ell = 3$.

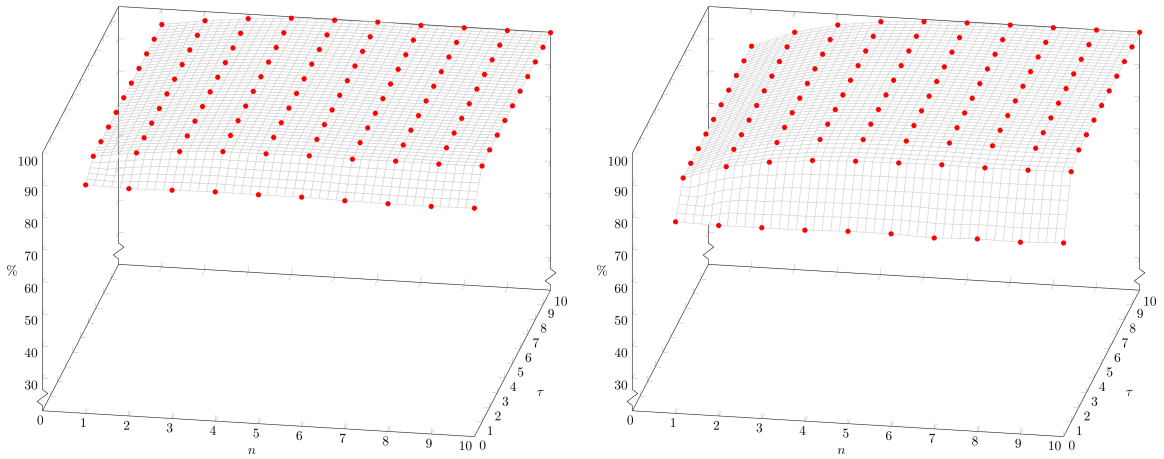
(c) $\ell = 4$. (d) $\ell = 5$.

Figure 5.2 – Variation on the percentage of τ-injective equivalence classes for $m = 5$ and several values of ℓ, n and τ.

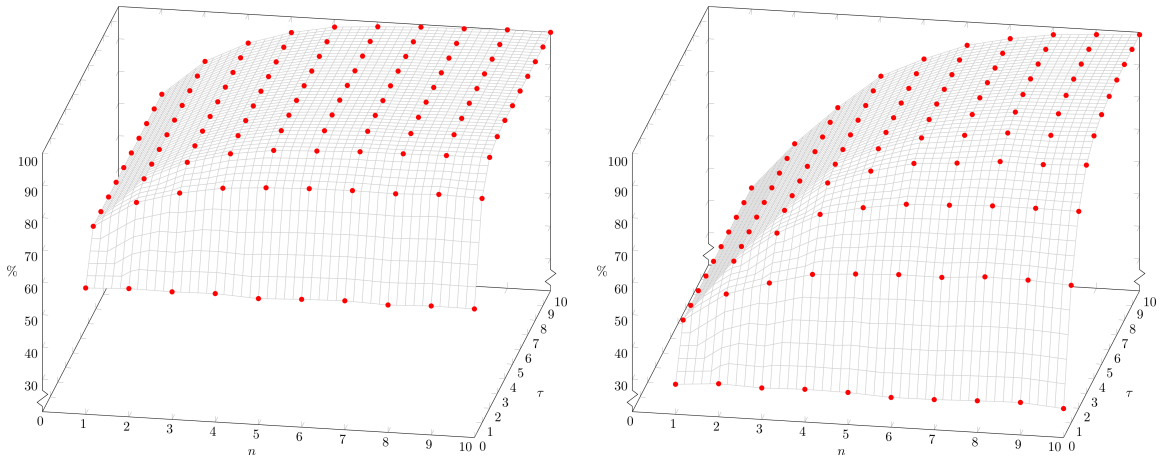
The results for $\ell = 3$ (Figures 5.2b and 5.3b), also show a significant growing of the values with n (it goes from values around 90% to values near 100%). A more careful observation of the percentages corresponding to $\tau = 10$, allow us to conclude that when $n \geq 3 = \ell$, the percentage of ω-injective LFTs is above 95%.

Observing all the figures, it can be noticed that the approximate percentage value, specially for low values of n, suffers a big reduction when ℓ increases from 2 to 5.

However, the growth, as a function of n, is much steeper for higher values of ℓ. This ensures that, for a not so large value of n, the percentage of ω-injective LFTs is very high. Therefore, if one uniformly random generates LFTs, it is highly probable to get ω-injective ones.



(a) $\ell = 2$. (b) $\ell = 3$.



(c) $\ell = 4$. (d) $\ell = 5$.

Figure 5.3 – Variation on the percentage of τ-injective equivalence classes for $m = 5$ and several values of ℓ, n and τ (from a different perspective than that from Figure 5.2).

We also show the results of an additional experiment done with $\ell = m = 8$, $n \in \{1, \ldots, 10\}$ and $\tau \in \{0, 1, \ldots, 10\}$. The percentages of τ-injective LFTs obtained are presented in Figure 5.4. Again, for values of n slightly larger than ℓ and m, one can see that the percentage of ω-injective LFTs is very high.

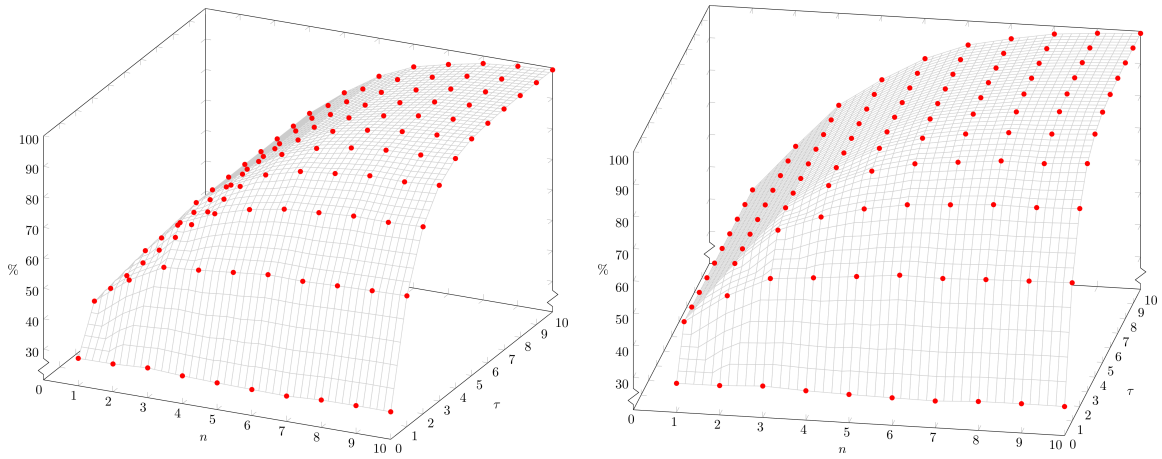


Figure 5.4 – Variation on the percentage of τ-injective equivalence classes for $\ell = 8$, $m = 8$, and several values of n and τ (from two different perspectives).

From all the experimental results presented we may draw two very important conclusions. First, the number of injective equivalence classes is very high and seems to grow exponentially as the structural parameters ℓ and n increase. This suggests that a brute force attack to the key space of a cryptographic system that uses these transducers is not feasible. Second, the percentage of equivalence classes of ω-injective LFTs, with structural parameters ℓ, m, n, is very high, for values of n slightly larger than ℓ and m. This lead us to believe that if one uniformly random generates an LFT, it is highly probable to find an injective one.

Chapter 6

Inverses of Linear Finite Transducers with Memory

In what follows, let $\mathbb{F}$ be a field, $\ell, m \in \mathbb{N}$, $\mathcal{X} = \mathbb{F}^{\ell}$, $\mathcal{Y} = \mathbb{F}^{m}$, and $\tau \in \mathbb{N}_0$.

6.1 Linear Finite Transducers with Memory

Given $h, k \in \mathbb{N}_0$ not simultaneously null, it is easy to see that a transducer, $M_\phi = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h \times \mathcal{Y}^k, \delta_\phi, \lambda_\phi \rangle$, with memory (h, k), in the sense of Definition 3.36, is linear if and only if the function ϕ can be expressed in the form

$$\phi(x_1, x_2, \ldots, x_h, x_{h+1}, y_1, \ldots, y_k) = \sum_{i=0}^{h} a_i x_{h+1-i} + \sum_{j=1}^{k} b_j y_{k+1-j}, \tag{6.1}$$

for some $a_0, \ldots, a_h \in \mathcal{M}_{m\times\ell}(\mathbb{F})$, $b_1, \ldots, b_k \in \mathcal{M}_m(\mathbb{F})$, and where $x_i \in \mathcal{X}$ for $i \in \{1, \ldots, h+1\}$, and $y_j \in \mathcal{Y}$ for $j \in \{1, \ldots, k\}$. If the function ϕ is not presented in the form (6.1), the construction of the matrices $a_0, \ldots, a_h \in \mathcal{M}_{m\times\ell}(\mathbb{F})$, and $b_1, \ldots, b_k \in \mathcal{M}_m(\mathbb{F})$, is similar to the construction of the structural matrices presented in Example 3.42. However, the usual way to define an LFT with memory is by presenting ϕ as an expression of the form (6.1). Nonetheless, the results and methods presented

in Chapter 3 can be easily applied, since the structural matrices of such an LFT, say A, B, C, D, are explicitly given in terms of the matrices $a_0, \ldots, a_h$, $b_1, \ldots, b_k$ as follows.

Let s be a state of M_ϕ, which is a vector of dimension $\ell h + km$ of the form

$$s = \begin{bmatrix} x_1 \\ \vdots \\ x_h \\ y_1 \\ \vdots \\ y_k \end{bmatrix},$$

where $x_i \in \mathcal{M}_{\ell\times 1}(\mathbb{F})$ for $i \in \{1, \ldots, h\}$, and $y_j \in \mathcal{M}_{m\times 1}(\mathbb{F})$ for $j \in \{1, \ldots, k\}$. Putting

$$C = \begin{bmatrix} a_h & \cdots & a_1 & b_k & \cdots & b_1 \end{bmatrix}, \tag{6.2}$$

and

$$D = a_0, \tag{6.3}$$

it follows that

$$\lambda_\phi(s, x_{h+1}) = \phi(x_1, \ldots, x_h, x_{h+1}, y_1, \ldots, y_k) = Cs + Dx_{h+1}.$$

Recalling that, by Definition 3.36,

$$\delta_\phi(< x_1, \ldots, x_h, y_1, \ldots, y_k >, x) =< x_2, \ldots, x_h, x, y_2, \ldots, y_k, y >,$$

where $y = \phi(x_1, \ldots, x_h, x, y_1, \ldots, y_k)$, if one takes

$$B = \left[\begin{array}{c} B_1 \\ \hdashline B_2 \end{array}\right] = \left[\begin{array}{c} 0_{(h-1)\ell\times\ell} \\ \hdashline I_\ell \\ \hdashline 0_{(k-1)m\times\ell} \\ a_0 \end{array}\right], \tag{6.4}$$

and

$$A = \left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right] = \left[\begin{array}{ccccc:ccccc} 0_\ell & I_\ell & & & & & & & & \\ & 0_\ell & I_\ell & & & & & & & \\ & & \ddots & \ddots & & & & 0_{h\ell\times km} & & \\ & & & 0_\ell & I_\ell & & & & & \\ 0_\ell & 0_\ell & \cdots & 0_\ell & 0_\ell & & & & & \\ \hdashline & & & & & 0_m & I_m & & & \\ & & & & & & 0_m & I_m & & \\ & 0_{(k-1)m\times h\ell} & & & & & & \ddots & \ddots & \\ & & & & & & & & 0_m & I_m \\ a_h & a_{h-1} & \cdots & a_2 & a_1 & b_k & b_{k-1} & \cdots & b_2 & b_1 \end{array}\right], \tag{6.5}$$

it can easily be seen that

$$\delta_\phi(s, x) = As + Bx.$$

Therefore, the structural matrices of M_ϕ are constructed from the matrices $a_0, \ldots, a_h$, $b_1, \ldots, b_k$ as in equations (6.2)–(6.5). Notice that a number of rows or columns lesser than 1 in $0_{i\times j}$ denotes the empty matrix.

Example 6.1. *Consider the transducer* $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, (\mathbb{F}_2^2)^2 \times \mathbb{F}_2^3, \delta, \lambda\rangle$ *defined in Example 3.38. Recall that* M *is the LFT with memory* $(2, 1)$ *defined by*

$$y_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} x_{t-2} + y_{t-1}, \text{ for } t \geq 0,$$

where $x_i \in \mathbb{F}_2^2$, *for* $i \geq -2$, $y_j \in \mathbb{F}_2^3$, *for* $j \geq -1$, *and* $< x_{-2}, x_{-1}, y_{-1} >$ *is the initial state of the transducer. That is,* M *is defined by an expression of the form* (6.1)*:*

$$y_t = a_0x_t + a_1x_{t-1} + a_2x_{t-2} + b_1y_{t-1}, \text{ for } t \geq 0,$$

where

$$a_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad a_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \text{ and } b_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, the structural matrices of M are

$$A = \left[\begin{array}{cc:c} 0_2 & I_2 & \\ 0_2 & 0_2 & 0_{4\times 3} \\ \hdashline a_2 & a_1 & b_1 \end{array}\right] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \left[\begin{array}{c} 0_2 \\ \hdashline I_2 \\ \hdashline a_0 \end{array}\right] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} a_2 & a_1 & b_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } D = a_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Now, we can easily compute $\mathrm{rank}(\Delta_M)$*, which is equal to 4, and, since* $\mathrm{size}(M) = 7$*, conclude that M is not minimal.*

6.2 Injectivity of LFTs with Memory

From Theorem 3.40, one already knows that the study of injectivity of LFTs with memory can be reduced to the study of LFTs with only input memory. More precisely,

an LFT with memory (h,k), $M_\varphi = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h \times \mathcal{Y}^k, \delta_\varphi, \lambda_\varphi \rangle$, defined by

$$\varphi(x_1, x_2, \ldots, x_h, x_{h+1}, y_1, \ldots, y_k) = \sum_{i=0}^{h} a_i x_{h+1-i} + \sum_{j=1}^{k} b_j y_{k+1-j},$$

is τ-injective if and only if the LFT with input memory $(h,0)$, $M_{\bar{\varphi}} = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h, \delta_{\bar{\varphi}}, \lambda_{\bar{\varphi}} \rangle$, defined by

$$\bar{\varphi}(x_1, x_2, \ldots, x_h, x_{h+1}) = \sum_{i=0}^{h} a_i x_{h+1-i},$$

is τ-injective. We say that $M_{\bar{\varphi}}$, as defined above, is the *input memory LFT corresponding* to M_φ.

From last chapter, one also knows that the transfer function matrix of an LFT can be used to check τ-injectivity. Now, we show how to quickly get that matrix for an LFT with input memory, which, from the observation made above, simplifies the process of checking injectivity for both LFTs with input memory and LFTs with memory in general.

Let Γ be the set of all linear maps from $\mathcal{X}^{h+1}$ to $\mathcal{Y}$, for all $h \in \mathbb{N}_0$, which can be given by linear forms $\sum_{i=0}^{h} a_i x_{h-i}$. Note that, necessarily, $a_i \in \mathcal{M}_{m\times\ell}(\mathbb{F})$, and $x_i \in \mathbb{F}^\ell$. Linear finite transducers with input memory are exactly the ones defined by functions in Γ, and this set can be identified with $\mathcal{M}_{m\times\ell}(\mathbb{F}[z]) \simeq \mathcal{M}_{m\times\ell}(\mathbb{F})[z]$ through the map $\psi : \Gamma \to \mathcal{M}_{m\times\ell}(\mathbb{F}[z])$ defined by

$$\psi\left(\sum_{i=0}^{h} a_i x_{h-i}\right) = \sum_{i=0}^{h} a_i z^i,$$

which is clearly a bijection. Thus, in what follows, we will use indistinctly either the linear form $L = \sum_{i=0}^{h} a_i x_{h-i}$ or the corresponding polynomial matrix $\psi(L)$ to represent the LFT with input memory defined by them.

Let M be an LFT with input memory $(h,0)$, defined by $\sum_{i=0}^{h} a_i x_{h-i} \in \Gamma$. Since the

structural matrices of M are

$$A = \begin{bmatrix} 0_\ell & I_\ell & & & \\ & 0_\ell & I_\ell & & \\ & & \ddots & \ddots & \\ & & & 0_\ell & I_\ell \\ 0_\ell & 0_\ell & \cdots & 0_\ell & 0_\ell \end{bmatrix}, \qquad B = \begin{bmatrix} 0_{(h-1)\ell \times \ell} \\ I_\ell \end{bmatrix},$$

$$C = \begin{bmatrix} a_h & \cdots & a_1 & b_k & \cdots & b_1 \end{bmatrix}, \qquad D = \begin{bmatrix} a_0 \end{bmatrix},$$

then

$$I - Az = \begin{bmatrix} I_\ell & -zI_\ell & & & \\ & I_\ell & -zI_\ell & & \\ & & \ddots & \ddots & \\ & & & I_\ell & -zI_\ell \\ & & & & I_\ell \end{bmatrix},$$

and

$$(I - Az)^{-1} = \begin{bmatrix} I_\ell & zI_\ell & z^2 I_\ell & \cdots & z^{h-1} I_\ell \\ & I_\ell & zI_\ell & & z^{h-2} I_\ell \\ & & \ddots & \ddots & \vdots \\ & & & I_\ell & zI_\ell \\ & & & & I_\ell \end{bmatrix}.$$

Consequently, the transfer function matrix of M is

$$H = C(I - Az)^{-1} Bz + D \;=\; C \begin{bmatrix} z^h I_\ell \\ \vdots \\ z^2 I_\ell \\ zI_\ell \end{bmatrix} + D \;=\; \sum_{i=0}^{h} a_i z^i \;=\; \psi\left(\sum_{i=0}^{h} a_i x_{h-i} \right).$$

We just proved the following proposition.

Proposition 6.2. *Let $h \in \mathbb{N}$, and let M be a linear finite transducer with input memory $(h, 0)$, defined by $\sum_{i=0}^{h} a_i x_{h-i} \in \Gamma$. Then, the transfer function matrix of M is*

$$H = \psi\left(\sum_{i=0}^{h} a_i x_{h-i}\right).$$

Example 6.3. *Let $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^2, (\mathbb{F}_2^2)^2, \delta, \lambda \rangle$ be the LFT with input memory $(2, 0)$ defined by*

$$y_t = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} x_{t-2}, \text{ for } t \geq 0,$$

where $x_i \in \mathbb{F}_2^2$, for $i \geq -2$, $y_j \in \mathbb{F}_2^2$, for $j \geq 0$, and $< x_{-2}, x_{-1} >$ is the initial state of the transducer. The transfer function matrix of M is

$$\begin{aligned} H(z) &= \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} z + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} z^2 \\ &= \begin{bmatrix} z + z^2 & 1 + z + z^2 \\ z + z^2 & 1 + z + z^2 \end{bmatrix}. \end{aligned}$$

Since $\det(H) = 0$, from Corollary 5.3, it follows that, for any $\tau \in \mathbb{N}_0$, M is not τ-injective.

Example 6.4. *Let $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, (\mathbb{F}_2^2)^2 \times (\mathbb{F}_2^3)^2, \delta, \lambda \rangle$ be the LFT with memory $(2, 2)$ defined by*

$$y_t = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} x_{t-2} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} y_{t-2}, , \text{ for } t \geq 0,$$

where $x_i \in \mathbb{F}_2^2$, for $i \geq -2$, $y_j \in \mathbb{F}_2^3$, for $j \geq -2$, and $< x_{-2}, x_{-1}, y_{-2}, y_{-1} >$ is the initial state of the transducer. The transfer function matrix of the corresponding input memory LFT is

$$H = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} z + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} z^2$$
$$= \begin{bmatrix} z^2 & 1 \\ 0 & 1+z+z^2 \\ z^2 & 1+z \end{bmatrix}.$$

Since $\operatorname{rank}(H) = 2$*, it follows, from Corollary 5.3, that* M *is* τ*-injective for some* $\tau \in \mathbb{N}_0$*. The Smith normal form of* H *is*

$$\begin{bmatrix} 1 & 0 \\ 0 & z^2 \\ 0 & 0 \end{bmatrix}.$$

Therefore, from Theorem 5.2, $\tau = 2$ *is the least delay* $\tau \in \mathbb{N}_0$ *such that* M *is* τ*-injective.*

6.3 Post-Initial Linear Transducers

Let $V = \mathcal{M}_{m\times\ell}(\mathbb{F})$, and $R = \mathcal{M}_m(\mathbb{F})$. In what follows we will regard $\mathcal{X}$ as left V-module, and $\mathcal{Y}$ as left R-module. Consider the family, Θ, of maps $\theta : \mathcal{X}^\omega \to \mathcal{Y}^\omega$ given by

$$y_t = \sum_{i=1}^{\eta} \left(\alpha_{t,i-1}\, x_{t+1-i} + \beta_{t,i}\, y_{t-i}\right), \text{ for } t \geq 0, \tag{6.6}$$

where $\eta \in \mathbb{N}$, $\alpha_{t,i-1} \in V, \beta_{t,i} \in R$, and

$$\forall t \geq i-1, \quad \alpha_{t,i-1} = a_{i-1} \quad \text{and} \quad \forall t \geq i, \quad \beta_{t,i} = b_i, \tag{6.7}$$

with $a_{i-1} \in V$, $b_i \in R$, for $i \in \{1, \ldots, \eta\}$. The variables with negative indices are free and a map in Θ is determined by their values, which one should think of as a set of

initial values. The family Θ is determined by the array of constants (its coefficients) presented in Table 6.1.

		Input Coefficients (ICs)					Output Coefficients (OCs)				
		i					i				
		1	2	3	$\cdots$	η	1	2	3	$\cdots$	η
t	0	a_0	$\alpha_{0,1}$	$\alpha_{0,2}$	$\cdots$	$\alpha_{0,\eta-1}$	$\beta_{0,1}$	$\beta_{0,2}$	$\beta_{0,3}$	$\cdots$	$\beta_{0,\eta}$
	1	a_0	a_1	$\alpha_{1,2}$	$\cdots$	$\alpha_{1,\eta-1}$	b_1	$\beta_{1,2}$	$\beta_{1,3}$	$\cdots$	$\beta_{1,\eta}$
	2	a_0	a_1	a_2	$\cdots$	$\alpha_{2,\eta-1}$	b_1	b_2	$\beta_{2,3}$	$\cdots$	$\beta_{2,\eta}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$\eta-1$	a_0	a_1	a_2	$\cdots$	$a_{\eta-1}$	b_1	b_2	b_3	$\cdots$	$\beta_{\eta-1,\eta}$
	$\geq \eta$	a_0	a_1	a_2	$\cdots$	$a_{\eta-1}$	b_1	b_2	b_3	$\cdots$	b_η

Table 6.1 – Coefficients of Θ.

When defining such a family Θ, at this point, we can give either a set of equations as in (6.6) or a table of coefficients as in Table 6.1.

Example 6.5. *Let $\mathcal{X} = \mathcal{Y} = \mathbb{F}_3$. Consider the family of maps $\theta : \mathcal{X}^\omega \to \mathcal{Y}^\omega$ given by*

$$\begin{cases} y_0 &= x_0 + x_{-1} + 2y_{-1}; \\ y_t &= x_t + x_{t-1} + y_{t-1} + y_{t-2}, \quad \textit{for } t \geq 1; \end{cases}$$

where $< x_{-1}, y_{-2}, y_{-1} >$ is the set of initial values. This family of maps can also be defined by the following table.

		ICs		*OCs*	
		i		i	
		1	2	1	2
t	0	1	1	2	0
	≥ 1	1	1	1	1

Example 6.6. *Let $\mathcal{X} = \mathbb{F}_2^2$ and $\mathcal{Y} = \mathbb{F}_2^3$. Consider the family of maps $\theta : \mathcal{X}^\omega \to \mathcal{Y}^\omega$ given by the coefficients in the following table.*

		ICs			*OCs*		
		i			i		
		1	2	3	1	2	3
t	0	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
	1	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
	≥ 2	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Then, the family of maps can also be given by the set of equations

$$\begin{cases} y_0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} x_0 + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} x_{-2} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_{-3}, \\ y_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} x_1 + \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} x_0 + \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x_{-1}, \\ y_t = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} x_{t-1} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x_{t-2} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} y_{t-3}, \ \textit{for } t \geq 2, \end{cases}$$

where $< x_{-2}, x_{-1}, y_{-3}, y_{-2}, y_{-1} >$ *is the set of initial values.*

For any given set of initial values, the corresponding map θ is a linear affine map of vector spaces over $\mathbb{F}$, and in the case they are all zero it is, of course, linear. Also, the fact that the sequences $(\alpha_{t,i})_t$ and $(\beta_{t,i})_t$ are eventually constant implies that θ is, what

Nerode calls, an *automaton transformation*, *i.e.*, is induced by a finite transducer, by a straightforward generalisation of [Ner58, Lemma 3] to our setting. We note that this result still holds in the general case of arbitrary initial values, since one can still use the same argument as in [Ner58, Lemma 3] to show that θ has a finite number of what Nerode calls *intrinsic states*, and then [Ner58, Lemma 2] applies. These initial values can also be thought of as states of the transducer, using a construction completely analogous to Tao's transducer with memory [Tao09].

All of the above shows that the following definition makes sense.

Definition 6.7. *A* post-initial linear transducer *(PILT) is a transducer induced by a recurrence relation as in* (6.6)*. If h is the largest value of $i \in \{1, \dots, \eta\}$ such that $\alpha_{t,i-1} \neq 0, \forall t \leq i-1$, and k is the largest value of $j \in \{1, \dots, \eta\}$ such that $\beta_{t,j} \neq 0, \forall t \leq j$, then one calls the corresponding transducer a PILT with memory (h, k), and $S = \mathcal{X}^h \times \mathcal{Y}^k$ is its set of states.*

Observation: If one represents a PILT with order (h, k) by a table similar to table 6.1, then h is the index minus 1 of the highest column containing the input coefficients that has a non-zero entry. And k is the index of the highest column containing the output coefficients that has a non-zero entry. Of course, the linear finite transducers with memory defined in the previous section correspond to the special case where the sequences $(\alpha_{t,i})_t$ and $(\beta_{t,i})_t$ are constant.

Example 6.8. *Let M be the PILT induced by the recurrence relation of Example 6.5. Then, M is a PILT with memory* $(1, 2)$*. And, taking, for example, $s =< 1, 2, 0 >$, one has*

$$\lambda(s, 11201) = 21001.$$

Example 6.9. *Let M be the PILT induced by the recurrence relation of Example 6.6. Then, M is a PILT with memory* $(2, 3)$*.*

Recall that $\mathcal{X} = \mathbb{F}^\ell$, $\mathcal{Y} = \mathbb{F}^m$, and let $S = \mathcal{X}^{\eta-1} \times \mathcal{Y}^\eta$. Put $X(z) = \sum_{t \geq 0} x_t z^t \in \mathbb{F}^\ell[[z]] \simeq \mathbb{F}[[z]]^\ell$ and $Y(z) = \sum_{t \geq 0} y_t z^t \in \mathbb{F}^m[[z]] \simeq \mathbb{F}[[z]]^m$. Multiplying (6.6) by z^t and

adding for all $t \geq 0$, one obtains

$$\sum_{t\geq 0} y_t\, z^t = \sum_{t\geq 0}\sum_{i=1}^{\eta} \alpha_{t,i-1}\; x_{t+1-i} z^t + \sum_{t\geq 0}\sum_{i=1}^{\eta} \beta_{t,i} y_{t-i} z^t,$$

which is equivalent to

$$\begin{aligned}
Y(z) &= \sum_{i=1}^{\eta} z^{i-1} \left(\sum_{t\geq 0} \alpha_{t,i-1}\, x_{t+1-i}\, z^{t+1-i} \right) + \sum_{i=1}^{\eta} z^i \left(\sum_{t\geq 0} \beta_{t,i}\, y_{t-i}\, z^{t-i} \right) \\
&= \sum_{i=1}^{\eta} z^{i-1} \left(\left(\sum_{t=0}^{i-2} \alpha_{t,i-1}\, x_{t+1-i}\, z^{t+1-i} \right) + \left(\sum_{t\geq i-1} a_{i-1}\, x_{t+1-i}\, z^{t+1-i} \right) \right) + \\
&\quad + \sum_{i=1}^{\eta} z^i \left(\left(\sum_{t=0}^{i-1} \beta_{t,i}\, y_{t-i}\, z^{t-i} \right) + \left(\sum_{t\geq i} b_i\, y_{t-i}\, z^{t-i} \right) \right), \text{ from 6.7} \\
&= \sum_{i=1}^{\eta} a_{i-1}\, z^{i-1}\, X(z) + \sum_{i=1}^{\eta} b_i\, z^i\, Y(z) + \sum_{i=2}^{\eta}\sum_{t=0}^{i-2} \alpha_{t,i-1}\, x_{t+1-i}\, z^t + \\
&\quad + \sum_{i=1}^{\eta}\sum_{t=0}^{i-1} \beta_{t,i}\, y_{t-i}\, z^t.
\end{aligned}$$

Since

$$\sum_{i=k}^{\eta}\sum_{j=0}^{i-k} f(i,j) = \sum_{j=0}^{\eta-k}\sum_{i=j+k}^{\eta} f(i,j),$$

for $k \leq n$ (see Appendix B for a sketch of the proof), it follows that

$$g(z)Y(z) - f(z)X(z) = r(s), \tag{6.8}$$

where $g(z) = I - \sum_{i=1}^{\eta} b_i z^i \in \mathcal{P}_{\eta+1}(R[z])$, $f(z) = \sum_{i=0}^{\eta-1} a_i\, z^i \in \mathcal{P}_{\eta}(V[z])$, and $r : S \to \mathcal{P}_{\eta}(\mathbb{F}[z]^m)$ is given by:

$$r(s) = \sum_{t=0}^{\eta-1} \left(\sum_{i=t+2}^{\eta} \alpha_{t,i-1} x_{t+1-i} + \sum_{i=t+1}^{\eta} \beta_{t,i} y_{t-i} \right) z^t, \tag{6.9}$$

if $s =< x_{-(\eta-1)}, \ldots, x_{-1}, y_{-\eta}, \ldots, y_{-1} >$. We will say that s gives *the initial conditions*, or the *initial state*.

It is clear that the two forms of inducing a transducer, either by an equation of the form (6.6) or by one of the form (6.8), are equivalent.

Example 6.10. *Let M be the PILT with memory $(2,3)$ induced by the recurrence relation of Example 6.6. Then, M can also be defined by the equation*

$$g(z)Y(z) - f(z)X(z) = r(s),$$

with

$$\begin{aligned}
f(z) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} z + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} z^2, \\
g(z) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} z^3, \\
r(s) &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} x_{-2} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_{-3} + \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} x_{-1}z + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} y_{-1}z^2,
\end{aligned}$$

and $s =< x_{-2}, x_{-1}, y_{-3}, y_{-2}, y_{-1} >$.

We are now ready to state a result that will allow us to give a complete characterisation of left invertibility in PILTs, and consequently of LFTs with memory.

Proposition 6.11. *Let $f \in \mathcal{M}_{m\times\ell}(\mathbb{F})[z]$, $g \in \mathcal{M}_m(\mathbb{F})[z]$ with $g(0) = I$, and let $r : S \to \mathbb{F}[z]^m$ be given by an expression of the form (6.9). Now, let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be a PILT induced by the equation $gY - fX = r(s)$, as described above. Then, the series of inputs and outputs of M, for some initial conditions s, satisfy an equation of the form*

$$uX - vY = q,$$

for some $u \in \mathcal{M}_\ell(\mathbb{F})[z]$ *with* $u \equiv z^\tau I \pmod{z^{\tau+1}}$, $v \in \mathcal{M}_{\ell\times m}(\mathbb{F})[z]$, *and* $q \in \mathbb{F}[z]^\ell$, *if and only if*

$$\exists\, p \in \mathcal{M}_{\ell\times m}(\mathbb{F})[z] :\ pf \equiv z^\tau I \pmod{z^{\tau+1}}.$$

Proof. One direction is obvious. If there exists $p \in \mathcal{M}_{\ell\times m}(\mathbb{F})[z]$ such that $pf \equiv z^\tau I \pmod{z^{\tau+1}}$, then just by multiplying both sides of equation $gY - fX = r(s)$ by p, on the left, one immediately gets the desired result.

To prove the other direction, assume that there are u, v, q in the conditions described in the statement of the theorem. Since $u \equiv z^\tau I \pmod{z^{\tau+1}}$, there is a polynomial w, such that $u = z^\tau w$ and $w(0) = I$. Since $g(0) = I$, g is invertible in $\mathcal{M}_m(\mathbb{F})[[z]]$, and from $gY - fX = r(s)$, it follows that

$$Y = g^{-1}fX + g^{-1}r(s).$$

Substituting u and Y, in $uX - vY = q$, by the above expressions, one gets

$$\left(z^\tau w - vg^{-1}f\right) X = vg^{-1}r(s) + q.$$

Since this must be true for all $X \in \mathcal{X}^\omega \simeq \mathbb{F}[[z]]^\ell$, it follows that $vg^{-1}r(s) + q = 0$ and, consequently, $z^\tau w - vg^{-1}f$ must be the zero matrix, which then implies that

$$z^\tau I = w^{-1}vg^{-1}f,$$

where I is the identity matrix of the appropriate size. Moreover, since f and $z^\tau I$ are polynomials, one concludes that $w^{-1}vg^{-1}$ is also a polynomial, more precisely, an element of $\mathcal{M}_{\ell\times m}(\mathbb{F})[z]$. Therefore, making $p = w^{-1}vg^{-1}$, one gets the claimed result. □

We are now ready to give the characterisation of left invertible PILTs.

Theorem 6.12. *Let* M *be a PILT induced by* $f \in \mathcal{M}_{m\times\ell}(\mathbb{F})[z]$, $g \in \mathcal{M}_m(\mathbb{F})[z]$ *with* $g(0) = I$, *and* $r : S \to \mathbb{F}[z]^m$, *as before. Then,* M *has a left inverse with delay* τ *if*

and only if

$$\exists\, p \in \mathcal{M}_{\ell\times m}(\mathbb{F})[z] :\ pf \equiv z^\tau I \pmod{z^{\tau+1}}.$$

In that case, if $w \in \mathcal{M}_\ell(\mathbb{F})[z]$ *is such that* $pf = z^\tau w$*, with* $w(0) = I$*, then an inverse with delay* τ *of* M *is the transducer induced by*

$$wY - pgX = r'(s'),$$

where $r'(s')$ *is obtained by switching* x *and* y *in* $-pr(s)$*.*

Proof. Suppose M has a left inverse with delay τ, $M' = \langle \mathcal{Y}, \mathcal{X}, S', \delta', \lambda' \rangle$. Let $wY - vX = r'(s')$, with $w(0) = I$, be an equation that induces M'. Then, for any input-output pair (X, Y) of M, and for any initial conditions s, there are initial conditions s' of M' and a polynomial $\gamma \in \mathcal{P}_\tau(\mathbb{F}[z]^\ell)$ such that $(Y, z^\tau X + \gamma)$ is an input-output pair of M'. This implies that

$$wz^\tau X - vY = r'(s') - w\gamma,$$

and the previous proposition then applies.

Conversely, assume the existence of p as stated, and let u be such that $pf = z^\tau u$. Then $u(0) = I$, and multiplying by p the equation defining M, one gets:

$$pgY - pfX = pr(s) \ \Leftrightarrow\ u\,(z^\tau X) - pgY = -pr(s), \tag{6.10}$$

where $-pr(s)$ can be seen as an expression of the form (6.9), by introducing new variables with zero coefficients, if necessary. More precisely, if $\deg(p) = \rho$, then $-pr(s)$ is of the form

$$\sum_{t=0}^{\rho+\eta-1} \left(\sum_{i=t+2}^{\rho+\eta} \alpha'_{t,i-1} x_{t+1-i} + \sum_{i=t+1}^{\rho+\eta} \beta'_{t,i} y_{t-i} \right) z^t, \tag{6.11}$$

where $x_{-\eta}, \ldots, x_{-(\rho+\eta-1)}$, and $y_{-(\eta+1)}, \ldots, y_{-(\rho+\eta)}$ are the new variables, whose coefficients, in expression 6.11, are zero. Let $r'(s') : S' \to \mathcal{P}_{\rho+\eta}(\mathbb{F}[z]^l)$ be given by the expression obtained by switching x and y in (6.11), where $S' = \mathcal{Y}^{\rho+\eta} \times \mathcal{X}^{\rho+\eta-1}$ and

$$s' =< y_{-(\rho+\eta)}, \ldots, y_{-1}, x_{-(\rho+\eta-1)}, \ldots, x_{-1} > .$$

Since equation (6.10) is verified for any input-output pair (X, Y) of M, one concludes that the transducer M' induced by $uY - pgX = r'(s')$ is a left inverse of M with delay τ, and, for all $x_{-(\rho+\eta-1)}, \ldots, x_{-\eta} \in \mathcal{X}$, $y_{-(\rho+\eta)}, \ldots, y_{-(\eta+1)} \in \mathcal{Y}$,

$$s' =< y_{-(\rho+\eta)}, \ldots, y_{-1}, x_{-(\rho+\eta-1)}, \ldots, x_{-1} >$$

is an inverse state with delay τ of $s =< x_{-(\eta-1)}, \ldots, x_{-1}, y_{-\eta}, \ldots, y_{-1} >$. □

Note that the left inverse whose existence is here shown outputs a number of leading zeros before starting to recover the input. Furthermore, given p in the conditions of the theorem, we can easily find that inverse, by the last statement of the theorem. To find such a p, one can use the techniques in the proof of the following result, in which $\mathcal{M}(R)$ will denote the union of all rings of matrices over the ring R.

Theorem 6.13. *Let $F \in \mathcal{M}(\mathbb{F}[z])$. Then*

$$\left(\exists\, P \in \mathcal{M}(\mathbb{F}[z]) :\ PF \equiv z^\tau I \pmod{z^{\tau+1}}\right) \Leftrightarrow\ z^{\tau+1} \nmid d,$$

where d is the invariant factor with the highest degree of F in Smith's normal form, and I is the appropriate identity matrix.

Proof. Let $F \in \mathcal{M}(\mathbb{F}[z])$. Since $\mathbb{F}[z]$ is a principal ideal domain, there exist invertible matrices $U, V \in \mathcal{M}(\mathbb{F}[z])$, with the appropriate dimensions, and such that $D = [d_{i,j}] = UFV$ is the Smith's normal form of F. Recall that $d_{i,j} = 0$ for $i \neq j$, and $d_{i,i} \mid d_{j,j}$ for $i \leq j$. Then, one has

$$
\begin{aligned}
\exists\, P \in \mathcal{M}(\mathbb{F}[z]) :\ & PF \equiv z^{\tau} I \pmod{z^{\tau+1}} \Leftrightarrow \\
& \Leftrightarrow \exists\, P \in \mathcal{M}(\mathbb{F}[z]) :\ PU^{-1}UFV \equiv z^{\tau} V \pmod{z^{\tau+1}} \\
& \Leftrightarrow \exists\, P \in \mathcal{M}(\mathbb{F}[z]) :\ V^{-1}PU^{-1}D \equiv z^{\tau} I \pmod{z^{\tau+1}} \\
& \Leftrightarrow \exists\, P \in \mathcal{M}(\mathbb{F}[z]) :\ PD \equiv z^{\tau} I \pmod{z^{\tau+1}} \\
& \Leftrightarrow \exists\, P = [p_{i,j}] \in \mathcal{M}(\mathbb{F}[z]) :\ \begin{cases} p_{i,j} \equiv 0 \pmod{z^{\tau+1}}, & \text{if } i \neq j; \\ p_{i,i} d_{i,i} \equiv z^{\tau} \pmod{z^{\tau+1}}, & \text{otherwise.} \end{cases} \\
& \underset{(a)}{\Leftrightarrow} z^{\tau+1} \nmid d,
\end{aligned}
$$

where d is the invariant factor of F with the highest degree. The if part of (a) can be proven as follows. If $i \neq j$, just take $p_{i,j} = 0$. For the remaining case, since $z^{\tau+1} \nmid d$, there is a non-negative integer $k \leq \tau$ such that

$$d = c_k z^k + c_{k+1} z^{k+1} + \ldots,$$

for some $c_k, c_{k+1}, \ldots \in \mathbb{F}$ with $c_k \neq 0$. Therefore, if one takes $p = c_k^{-1} z^{\tau-k}$, one gets

$$pd \equiv z^{\tau} \pmod{z^{\tau+1}}.$$

Since $d_{i,i} \mid d_{j,j}$ when $i \leq j$, from $z^{\tau+1} \nmid d$ it follows that $z^{\tau+1} \nmid d_{i,i}$, for all i, and the same reasoning applies. □

From Proposition 6.11, Theorem 6.12 and Theorem 6.13 one gets the following necessary and sufficient condition for the left invertibility of PILTs.

Corollary 6.14. *Let $f \in \mathcal{M}_{m\times\ell}(\mathbb{F})[z]$, $g \in \mathcal{M}_m(\mathbb{F})[z]$ such that $g(0) = I$, and $r : S \to \mathbb{F}[z]^m$ is given by an expression of the form (6.9). Let $M = \langle \mathbb{F}^{\ell}, \mathbb{F}^m, S, \delta, \lambda \rangle$ be a PILT induced by the equation $gY - fX = r(s)$. Then, M is left invertible with delay*

τ if and only if

$$z^{\tau+1} \nmid d,$$

where d is the invariant factor with the highest degree of f, when f is seen as an element of $\mathcal{M}_{m\times\ell}(\mathbb{F}[z])$.

We can now state an algorithm to check τ-injectivity of PILTs, and to find a left inverse with delay τ, if it exists. Let $M = \langle \mathcal{X}, \mathcal{Y}, S, \delta, \lambda \rangle$ be the PILT induced by the equation $gY - fX = r(s)$, where $f \in \mathcal{M}_{m\times\ell}(\mathbb{F})[z]$, $g \in \mathcal{M}_m(\mathbb{F})[z]$ such that $g(0) = I$, and $r : S \to \mathbb{F}[z]^m$ given by an expression of the form (6.9).

1. Compute the Smith normal form of F, $D = [d_{i,j}]$, where F is the polynomial matrix corresponding to f. If the invariant factor with the highest degree is not a multiple of $z^{\tau+1}$, then the PILT is τ-injective and we should proceed to step 2. Otherwise, we should stop because the transducer is not τ-injective, and therefore there is no left inverse with delay τ of M.

2. Compute the matrices $U \in \mathcal{M}_m(\mathbb{F}[z])$ and $V \in \mathcal{M}_\ell(\mathbb{F}[z])$ such that $UFV = D$ (in fact these matrices are already computed in Step 1).

3. Construct a matrix $A = [a_{i,j}] \in \mathcal{M}_{\ell\times m}(\mathbb{F}[z])$ such that

$$\begin{cases} a_{i,j} \equiv 0 \pmod{z^{\tau+1}}, \text{ if } i \neq j; \\ a_{i,i}d_{i,i} \equiv z^\tau \pmod{z^{\tau+1}}, \text{ otherwise.} \end{cases} \tag{6.12}$$

4. Compute $P = VAU$.

5. Determine $W \in \mathcal{M}_\ell(\mathbb{F}[z])$ such that $PF = z^\tau W$, with $W(0) = I$, *i.e.*,

$$W = z^{-\tau}PF.$$

6. Compute PG, where G is the polynomial matrix corresponding to g. Let v be the matrix polynomial corresponding to PG.

7. Compute $pr(s)$, where p is the matrix polynomial corresponding to P.

Then, a left inverse with delay τ of M is the post-initial linear transducer M' induced by:

$$wY - vX = r'(s'),$$

where $r'(s')$ is obtained by switching x and y in $-pr(s)$.

Example 6.15. *Consider the PILT from Example 6.10. We will use the previous steps to show that M is 1-injective and to compute a left inverse with delay 1 of M.*

1. *Take*

$$F = \begin{bmatrix} z+z^2 & 1+z \\ z & z \\ z & 1+z \end{bmatrix}.$$

The Smith normal form of F is

$$D = \begin{bmatrix} 1 & 0 \\ 0 & z \\ 0 & 0 \end{bmatrix}.$$

Since $z^2 \nmid z$, it follows that the PILT is 1-injective.

2. *The matrices U, V such that $D = UFV$ are:*

$$U = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1+z & z \\ 1 & z+z^2 & 1+z^2 \end{bmatrix} \text{ and } V = \begin{bmatrix} 0 & 1 \\ 1 & z^2 \end{bmatrix}.$$

3. *Take, for example,*

$$A = \begin{bmatrix} z & z^2 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

which satisfies (6.12).

4.

$$P = VAU = \begin{bmatrix} 0 & 1+z & z \\ z & z & 0 \end{bmatrix}.$$

5.

$$W = z^{-1}PF = \begin{bmatrix} 1 & 0 \\ z^2 & 1 \end{bmatrix}$$

6. Let v be the matrix polynomial corresponding to

$$PG = \begin{bmatrix} 0 & 1+z+z^4 & z \\ z+z^4 & z+z^4 & 0 \end{bmatrix},$$

where G is the polynomial matrix corresponding to g, i.e.,

$$v = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} z^4.$$

7. Let p be the matrix polynomial corresponding to P. Then

$$pr(s) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_{-3} + \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x_{-2} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} y_{-3} \right) z + \\ + \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} x_{-1}z^2 + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} y_{-1}z^3.$$

A left inverse with delay 1 of M is thus the PILT $M' = \langle \mathbb{F}_2^3, \mathbb{F}_2^2, (\mathbb{F}_2^3)^4 \times (\mathbb{F}_2^2)^3, \delta', \lambda' \rangle$ induced by

$$wY - vX = r'(s'),$$

where w is the matrix polynomial corresponding to W, and $r'(s')$ is obtained by switch-

ing x and y in $-pr(s)$, i.e.,

$$r'(s') = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_{-3} + \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} y_{-2} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} x_{-3} \right) z + \\ + \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} y_{-1} z^2 + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} x_{-1} z^3.$$

Remark 6.16. *From the discussion in the proof of Theorem 6.12, the memory of a left inverse, constructed using our algorithm, is at most $(\eta+\rho, \eta-1+\rho)$, where $\rho = \deg(p)$. To ensure that ρ is not too large, roughly speaking, we can take P as the remainder of the division of VAU by $z^{\tau+1}$, instead of taking $P = VAU$ (in step 4.). In this way, we still have $pf \equiv z^\tau I \pmod{z^{\tau+1}}$, as required by Theorem 6.12, and $\deg(p) \leq \tau$. This change ensures that the memory of the left inverse is at most $(\eta+\tau, \eta-1+\tau)$. By a similar argument, it can be seen that, if the memory of the PILT is (h,k), then we can find a left inverse that has memory at most $(k+\tau, h+\tau)$.*

Example 6.17. *Let $M = \langle \mathbb{F}_2^2, \mathbb{F}_2^3, (\mathbb{F}_2^2)^2 \times \mathbb{F}_2^3, \delta, \lambda \rangle$ be the LFT with memory $(2,1)$ induced by the equation*

$$g(z)Y(z) - f(z)X(z) = r(s),$$

with

$$\begin{aligned} f(z) &= \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} z + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} z^2, \\ g(z) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} z, \\ r(s) &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} x_{-1} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} x_{-2} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} y_{-1} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} x_{-1} z. \end{aligned}$$

We prove that M is 2-injective and compute a left inverse with delay 2 of M.

1. *Take*
$$F = \begin{bmatrix} z^2 & 1 \\ 0 & 1+z+z^2 \\ z^2 & 1+z \end{bmatrix}.$$
The Smith normal form of F is
$$D = \begin{bmatrix} 1 & 0 \\ 0 & z^2 \\ 0 & 0 \end{bmatrix}.$$
Since $z^3 \nmid z^2$, it follows that the LFT is 2-injective.

2. *The matrices U, V such that $D = UFV$ are:*
$$U = \begin{bmatrix} 1 & 0 & 0 \\ z & 1 & 1+z \\ 1+z+z^2 & z & 1+z+z^2 \end{bmatrix} \text{ and } V = \begin{bmatrix} 0 & 1 \\ 1 & z^2 \end{bmatrix}.$$

3. *Take, for example,*
$$A = \begin{bmatrix} z^2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$
which satisfies (6.12).

4. *One has*
$$VAU = \begin{bmatrix} z & 1 & 1+z \\ z^2+z^3 & z^2 & z^2+z^3 \end{bmatrix}.$$
Take
$$P = \begin{bmatrix} z & 1 & 1+z \\ z^2 & z^2 & z^2 \end{bmatrix}.$$

5.
$$W = z^{-\tau}PF = z^{-2}\begin{bmatrix} z^2 & 0 \\ 0 & z^2+z^4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1+z^2 \end{bmatrix}.$$

6. *Let v be the matrix polynomial corresponding to*
$$PG = \begin{bmatrix} z^2 & 1+z & 1+z \\ z^2+z^3 & z^2+z^3 & z^2 \end{bmatrix},$$
where G is the polynomial matrix corresponding to g, i.e.,
$$v = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} z^2 + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} z^3.$$

7. *Let p be the matrix polynomial corresponding to P. Then*
$$pr(s) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} x_{-2} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_{-1} + \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} x_{-1} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} y_{-1} \right) z + \\ + \left(\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x_{-2} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} y_{-1} \right) z^2 + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x_{-1} z^3.$$

A left inverse with delay 2 of M is hence the PILT $M' = \langle \mathbb{F}_2^3, \mathbb{F}_2^2, (\mathbb{F}_2^3)^3 \times (\mathbb{F}_2^2)^4, \delta', \lambda' \rangle$ induced by
$$wY - vX = r'(s'),$$
where w is the matrix polynomial corresponding to W, and $r'(s')$ is obtained by switching x and y in $-pr(s)$, i.e.,
$$r'(s') = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} y_{-2} + \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_{-1} + \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} y_{-1} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_{-1} \right) z + \\ + \left(\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} y_{-2} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} x_{-1} \right) z^2 + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} y_{-1} z^3.$$

For example, take the following state of M,

$$s = \left\langle \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\rangle,$$

and the input sequence

$$\alpha = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then

$$\lambda(s, \alpha) = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

An inverse state with delay 2 of s is the state

$$s' = \left\langle \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\rangle,$$

and

$$\lambda'\left(s', \lambda(s, \alpha)\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

as expected.

Chapter 7

Conclusion

In this work we gave an unified presentation of the concepts and known results, as far as we could establish, on general linear finite transducers as well as on linear transducers with memory. We simplified the language used in previous works, by introducing a more categorical point of view, and contributed with a wide variety of examples to illustrate the concepts and techniques presented.

We improved the existing results about equivalence of LFTs, which are due to Tao, and this led us to a method to check the equivalence of LFTs. This method allowed us to compute the size of equivalence classes in $\mathcal{L}_n/\sim_n$, for $n \in \mathbb{N}$, by studying how the augmented diagnostic matrices of equivalent transducers in $\mathcal{L}_n$ vary. The results presented, as well as the techniques in their proofs, were used to present an algorithm that enumerates the LFTs in $[M]_{\sim_n}$, where M is an LFT of size $n \in \mathbb{N}$. We also introduced a notion of canonical LFT and proved that each equivalence class has exactly one of these transducers. A recurrence relation was then deduced to compute the number of canonical LFTs with the same size, which made possible to have a way to compute the number of non-equivalent LFTs.

Regarding the injectivity of LFTs, we recalled and proved two necessary and sufficient conditions, which are due to Zongduo and Dingfeng, for an LFT to be τ-injective. We then showed how to implement an algorithm that employs one of these conditions

to check τ-injectivity. Using uniform random generation of LFTs, and the previous results on the number and size of equivalence classes, we explained how to estimate the number and percentage of non-equivalent LFTs that are τ-injective ($\tau \in \mathbb{N}_0$). We also showed how these methods can be implemented in `Python` using some `Sage` modules to deal with matrices. Several experimental results were presented which strongly suggested two things. First, a brute force attack to the key space of a cryptographic system that uses these transducers is not feasible. Second, if one uniformly random generates an LFT, it is highly probable to find an injective one. Moreover, since the values obtained are really close to 100%, this is a good indicator that if one uniformly random generates an LFT with memory, which by definition satisfies the condition $n = h\ell + km$ (where $h, k \in \mathbb{N}$), then it is highly probable to get one that is ω-injective. However, it remains to study this particular case. In fact, as future work, it would be interesting to do a complete characterisation of LFTs with memory, and also do a study on the number and percentage of τ-injective LFTs with memory, analogous to the one we presented for LFTs. Such a study would complement the work here presented on the characterisation of LFTs for cryptographic purposes. Furthermore, using the results about the size of equivalence classes, it can be explored how to construct an uniform random generator of non-equivalent LFTs.

Despite the work already done, mainly by Tao, on the invertibility theory of finite transducers, an algorithm to compute left inverses of invertible LFTs with memory was never presented. Such an algorithm is of fundamental importance in the key generation process using random generation. By introducing an appropriate extension of the notion of LFT, that we called PILT, and working on rings of formal power series and some associated modules, we found an algorithm to compute left inverses of invertible LFTs with memory. We also gave a necessary and sufficient condition for the injectivity of these transducers. As future work in this subject, the new technique provided to invert LFTs with memory can be explored to deal with the invertibility of quasi-linear finite transducers over finite fields (as defined by Tao [Tao09]). This is the kind of non-linear FTs used in the known FAPKCs and, since the structure of LFTs

and quasi-LFTs are quite similar, we believe that such a study could be successful[1].

Another fundamental direction of research is the study of general non-linear finite transducers and their invertibility, pursuing new classes of cryptographic systems using transducers.

[1]Let $M = \langle \mathcal{X}, \mathcal{Y}, \mathcal{X}^h \times \mathcal{Y}^k, \delta, \lambda \rangle$ be a finite transducer with memory (h, k). M is said to be a $\tau-$*quasi-linear finite transducer* if is defined by an expression of the form $y_t = \sum_{i=0}^{\tau} a_i x_{t-i} + g(x_{t-\tau-1}, \ldots, x_{t-h}, y_{t-1}, \ldots, y_{t-k})$, where $g : \mathcal{X}^{h-\tau} \times \mathcal{Y}^k \to Y$ is a non-linear map.

Appendix A

Tables of Experimental Results

Below we present a set of tables with the estimates of the percentages obtained in the experiments described in Section 5.4.

		τ										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	**1**	90.88	95.21	95.21	95.21	95.21	95.21	95.21	95.21	95.21	95.21	95.21
	2	90.5	97.06	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2	97.2
	3	90.82	98.27	98.58	98.62	98.62	98.62	98.62	98.62	98.62	98.62	98.62
	4	91.1	99.07	99.53	99.57	99.57	99.57	99.57	99.57	99.57	99.57	99.57
	5	91.01	99.18	99.72	99.74	99.74	99.74	99.74	99.74	99.74	99.74	99.74
	6	91.07	99.37	99.92	99.95	99.96	99.96	99.96	99.96	99.96	99.96	99.96
	7	90.75	99.12	99.69	99.73	99.73	99.73	99.73	99.73	99.73	99.73	99.73
	8	90.64	99.31	99.76	99.81	99.81	99.81	99.81	99.81	99.81	99.81	99.81
	9	90.6	99.18	99.7	99.74	99.74	99.75	99.75	99.75	99.75	99.75	99.75
	10	90.85	99.39	99.85	99.89	99.89	99.89	99.89	99.89	99.89	99.89	99.89

Table A.1 – Estimates of the percentage of τ-injective equivalence classes for $\ell = 2$ and $m = 5$.

		τ										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	**1**	79.42	88.48	88.48	88.48	88.48	88.48	88.48	88.48	88.48	88.48	88.48
	2	79.08	92.77	93.61	93.61	93.61	93.61	93.61	93.61	93.61	93.61	93.61
	3	79.19	94.98	96.54	96.68	96.68	96.68	96.68	96.68	96.68	96.68	96.68
	4	79.22	96.31	98.27	98.47	98.48	98.48	98.48	98.48	98.48	98.48	98.48
	5	79.69	96.89	99.04	99.28	99.29	99.29	99.29	99.29	99.29	99.29	99.29
	6	79.68	97.14	99.39	99.66	99.70	99.71	99.71	99.71	99.71	99.71	99.71
	7	79.21	97.37	99.58	99.79	99.83	99.85	99.85	99.85	99.85	99.85	99.85
	8	79.72	97.22	99.52	99.79	99.82	99.82	99.82	99.82	99.82	99.82	99.82
	9	79.50	97.32	99.56	99.85	99.90	99.91	99.91	99.91	99.91	99.91	99.91
	10	80.07	97.64	99.83	100	100	100	100	100	100	100	100

Table A.2 – Estimates of the percentage of τ-injective equivalence classes for $\ell = 3$ and $m = 5$.

		τ										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	**1**	59.09	73.64	73.64	73.64	73.64	73.64	73.64	73.64	73.64	73.64	73.63
	2	59.70	81.83	84.60	84.60	84.60	84.60	84.60	84.60	84.60	84.60	84.60
	3	59.50	85.53	90.49	91.07	91.07	91.07	91.07	91.07	91.07	91.07	91.07
	4	59.76	87.83	93.95	95.01	95.13	95.13	95.13	95.13	95.13	95.13	95.13
	5	59.01	88.77	95.79	97.35	97.60	97.64	97.64	97.64	97.64	97.64	97.64
	6	59.58	89.29	96.39	98.14	98.48	98.52	98.53	98.53	98.53	98.53	98.53
	7	59.93	89.49	96.97	98.76	99.14	99.19	99.22	99.22	99.22	99.22	99.22
	8	59.43	89.30	97.14	98.87	99.35	99.49	99.51	99.51	99.51	99.51	99.51
	9	59.93	89.91	97.40	99.31	99.81	99.95	99.97	99.98	99.98	99.98	99.98
	10	59.81	89.46	97.64	99.51	99.99	100	100	100	100	100	100

Table A.3 – Estimates of the percentage of τ-injective equivalence classes for $\ell = 4$ and $m = 5$.

		τ										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	**1**	29.29	44.63	44.63	44.63	44.63	44.63	44.63	44.63	44.63	44.63	44.63
	2	30.26	53.48	59.11	59.11	59.11	59.11	59.11	59.11	59.11	59.11	59.11
	3	29.75	57.69	68.60	71.09	71.09	71.09	71.09	71.09	71.09	71.09	71.09
	4	30.13	61.15	75.19	80.37	81.63	81.63	81.63	81.63	81.63	81.63	81.63
	5	29.96	62.07	78.05	84.84	87.21	87.74	87.74	87.74	87.74	87.74	87.74
	6	29.21	62.69	79.92	88.01	91.37	92.52	92.79	92.79	92.79	92.79	92.79
	7	29.35	62.63	80.43	88.92	92.98	94.87	95.50	95.65	95.65	95.65	95.65
	8	29.78	63.60	81.02	90.20	94.50	96.43	97.33	97.62	97.67	97.67	97.67
	9	30.07	63.39	81.08	90.05	94.57	96.71	97.85	98.35	98.46	98.50	98.50
	10	28.97	62.58	80.92	90.70	95.22	97.24	98.34	98.87	99.14	99.25	99.26

Table A.4 – Estimates of the percentage of τ-injective equivalence classes for $\ell = 5$ and $m = 5$.

		τ										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
n	**1**	29.01	43.59	43.59	43.59	43.59	43.59	43.59	43.59	43.59	43.59	43.59
	2	29.11	52.44	57.91	57.91	57.91	57.91	57.91	57.91	57.91	57.91	57.91
	3	29.77	58.58	69.04	71.44	71.44	71.44	71.44	71.44	71.44	71.44	71.44
	4	29.11	59.60	73.92	79.13	80.16	80.16	80.16	80.16	80.16	80.16	80.16
	5	28.76	60.80	77.23	84.41	86.94	87.51	87.51	87.51	87.51	87.51	87.51
	6	28.52	62.01	79.32	87.49	90.88	92.30	92.55	92.55	92.55	92.55	92.55
	7	28.33	61.79	80.11	88.77	92.99	94.61	95.16	95.29	95.29	95.29	95.29
	8	28.98	62.25	80.95	89.98	94.20	96.11	97.09	97.47	97.55	97.55	97.55
	9	29.09	62.59	80.84	89.94	94.57	96.96	97.94	98.40	98.56	98.59	98.59
	10	29.01	62.86	81.34	90.75	95.36	97.63	98.56	99.06	99.28	99.34	99.35

Table A.5 – Estimates of the percentage of τ-injective equivalence classes for $\ell = 8$ and $m = 8$.

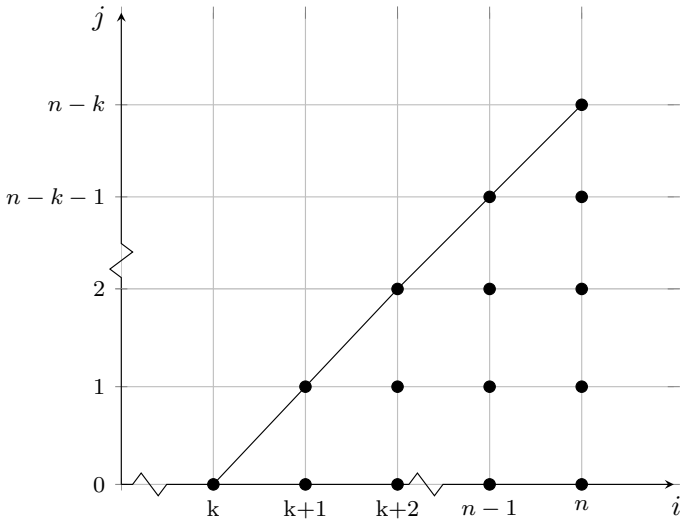
Appendix B

Change of Variables in Summations

Let $k, n \in \mathbb{N}$ such that $k \leq n$. Consider the sums

$$A = \sum_{i=k}^{n} \sum_{j=0}^{i-k} f(i,j), \quad \text{and} \quad B = \sum_{j=0}^{n-k} \sum_{i=j+k}^{n} f(i,j).$$

To prove that $A = B$, we just need to see that the sets of pairs of indices (i, j) in the summations A and B are the same. That is easily seen through the figure below, where those pairs of indices are represented. Notice that the equation of the line is $i = j + k \Leftrightarrow j = i - k$.



Bibliography

[Abu11] Sashad Abubaker. Probabilistic, Lightweight Cryptosystems based on Finite automata. Master's thesis, Departament of Computer Science, University of Victoria, 2011.

[AM69] Michael F. Atiyah and Ian G. Macdonald. *Introduction to Commutative Algebra*. Addison-Wesley Publishing Company, 1969.

[AMR12] Ivone Amorim, António Machiavelo, and Rogério Reis. Formal Power Series and the Invertibility of Finite Linear Transducers. In Rudolf Freund, Markus Holzer, Bianca Truthe, and Ulrich Ultes-Nitsche, editors, *Fourth Workshop on Non-Classical Models for Automata and Applications - NCMA 2012*, pages 33–48. Österreichische Computer Gesellschaft, 2012.

[AMR14a] Ivone Amorim, António Machiavelo, and Rogério Reis. Counting Equivalent Linear Finite Transducers Using a Canonical Form. In Markus Holzer and Martin Kutrib, editors, *Implementation and Application of Automata - 19th International Conference - CIAA 2014*, volume 8587 of *Lecture Notes in Computer Science*, pages 70–83. Springer, 2014.

[AMR14b] Ivone Amorim, António Machiavelo, and Rogério Reis. On the Invertibility of Finite Linear Transducers. *RAIRO - Theoretical Informatics and Applications*, 48(01):107–125, 2014.

[AMR14c] Ivone Amorim, António Machiavelo, and Rogério Reis. Statistical Study on the Number of Injective Linear Finite Transducers. In Suna Bensch,

Rudolf Freund, and Friedrich Otto, editors, *Sixth Workshop on Non-Classical Models for Automata and Applications - NCMA 2014*, pages 57–72. Österreichische Computer Gesellschaft, 2014.

[AMR15] Ivone Amorim, António Machiavelo, and Rogério Reis. On the Number of Linear Finite Transducers. *International Journal of Foundations of Computer Science*, 26(7):873–893, 2015.

[BI95] Feng Bao and Yoshihide Igarashi. Break Finite Automata Public Key Cryptosystem. In Zoltán Fülöp and Ferenc Gécseg, editors, *Automata, Languages and Programming*, volume 944 of *Lecture Notes in Computer Science*, pages 147–158. Springer Berlin Heidelberg, 1995.

[Dev15] The Sage Developers. *Sage Mathematics Software (Version 6.5)*, 2015. `http://www.sagemath.org`.

[Dif88] Whitfield Diffie. The First Ten Years of Public-Key Cryptography. *Proceedings of the IEEE*, 76(5):560–577, 1988.

[HZ99] Ou Haiwen and Dai Zongduo. Self-Injective Rings and Linear (Weak) Inverses of Linear Finite Automata over Rings. *SCIENCE CHINA Mathematics*, 42(2):140, 1999.

[Jac85] Nathan Jacobson. *Basic Algebra I*. W H Freeman & Co, 1985.

[McC71] Neal H. McCoy. *Introduction to Modern Algebra*. Allyn and Bacon, Boston, 1971.

[MP13] Gary L. Mullen and Daniel Panario. *Handbook of Finite Fields*. Chapman & Hall/CRC, 1st edition, 2013.

[MS68] James L. Massey and Michael K. Slain. Inverses of Linear Sequential Circuits. *IEEE Transactions on Computers*, C-17:330–337, April 1968.

[Ner58] Anil Nerode. Linear Automaton Transformations. *Proceedings of the American Mathematical Society*, 9(4):541–544, August 1958.

[New72] Morris Newman. *Integral Matrices*. Academic Press, 1972.

[Rut06] J. J. M. M. Rutten. Algebraic Specification and Coalgebraic Synthesis of Mealy Automata. *ENTCS*, 160(0):305–319, 2006.

[Sta72] Peter Starke. *Abstract Automata*. Academic Press, 1972.

[Tao73] Renji Tao. Invertible Linear Finite Automata. *Scientia Sinica*, XVI(4):565–581, November 1973.

[Tao88] Renji Tao. Invertibility of Linear Finite Automata Over a Ring. In Timo Lepistö and Arto Salomaa, editors, *Automata, Languages and Programming*, volume 317 of *Lecture Notes in Computer Science*, pages 489–501. Springer Berlin Heidelberg, 1988.

[Tao95a] Renji Tao. On Invertibility of Some Compound Finite Automata. Technical Report No. ISCAS-LCS-95-06, Laboratory for Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, 1995.

[Tao95b] Renji Tao. On R_a, R_b Transformation and Inversion of Compound Finite Automata. Technical Report No. ISCAS-LCS-95-10, Laboratory for Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, 1995.

[Tao09] Renji Tao. *Finite Automata and Application to Cryptography*. Springer Berlin Heidelberg, 2009.

[TC85] Renji Tao and Shihua Chen. A Finite Automaton Public Key Cryptosystem and Digital Signatures. *Chinese Journal of Computers*, 8(6):401–409, 1985. (in Chinese).

[TC86] Renji Tao and Shihua Chen. Two Varieties of Finite Automaton Public Key Cryptosystem and Digital Signatures. *Journal of Computer Science and Technology*, 1(1):9–18, 1986.

[TC97] Renji Tao and Shihua Chen. A Variant of the Public Key Cryptosystem FAPKC3. *Journal of Network and Computer Applications*, 20:283–303, July 1997.

[TC99] Renji Tao and Shihua Chen. The Generalization of Public Key Cryptosystem FAPKC4. *Chinese Science Bulletin*, 44(9):784–790, 1999.

[TCC97] Renji Tao, Shihua Chen, and Xuemei Chen. FAPKC3: A New Finite Automaton Public Key Cryptosystem. *Journal of Computer Science and Technology*, 12(4):289–305, July 1997.

[Val93] Robert J. Valenza. *Linear Algebra: An Introduction to Abstract Mathematics*. Springer New York, 1993.

[ZD96] Dai Zongduo and Ye Dingfeng. Weak Invertibility of Linear Finite Automata (I), Classification and Enumeration of Transfer Functions. *SCIENCE CHINA Mathematics*, 39(6):613, 1996.

[ZDL98] Dai Zongduo, Ye Dingfeng, and Kwokyan Lam. Weak Invertibility of Finite Automata and Cryptanalysis on FAPKC. In Kazho Ohta and Dingyi Pei, editors, *Advances in Cryptology–AsiaCrypt'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 227–241. Spring-Verlag, 1998.

Index

SNF, 25

ω-injective, 45

τ-injective, 45

FT, 33

LFT, 53

PID, 12

PILT, 113

alphabet, 33

basis, 15

bijection, 8

binary relation, 7

canonical LFT, 69

Cayley-Hamilton theorem, 26

characteristic polynomial, 26

congruence relation, 8

congruent modulo n, 8

diagnostic matrix, 56

divides, 10

empty word, 33

equivalence

- class, 7
- relation, 7

equivalent

- states, 38
- transducers, 40

field, 13

finite field, 13

finite transducer, 33, 34

- isomorphism, 37
- with input memory, 50
- with memory, 50

formal power series, 11

free response matrix, 85

function, 8

- bijective, 8
- image, 8
- injective, 8
- one-to-one, 8
- surjective, 8

Galois field, 13

group, 9

Abelian, 9
additive, 9
commutative, 9
multiplicative, 9
operation, 9

homomorphism, 27

ideal, 11
generated by S, 12
injective with delay τ, 45
input alphabet, 34
invariant factors, 25
inverse, 10
state with delay τ, 47
invertible, 10
isomorphic, 28, 37

left
inverse with delay τ, 49
invertible with delay τ, 49
module, 14
length, 33
letters, 33
linear
combination, 15
finite transducer, 53
isomorphism, 28
map, 27
transformation, 27
linearly
dependent, 15
independent, 15
localisation ring, 13

map, 8
mapping, 8
matrix, 16
adjoint, 21
column rank, 18
column space, 18
determinant, 20
entries, 17
equivalent, 24
identity, 17
inverse, 21
invertible, 21
left inverse, 22
left invertible, 22
maximal rank, 18
non-singular, 21
non-square, 16
rank, 18
reduced column echelon form, 22
reduced row echelon form, 22
right inverse, 22
right invertible, 22
row rank, 18
row space, 17
square, 16
transpose, 20
matrix of the linear application, 29
matrix polynomials, 19

minimal
    polynomial, 27
    transducer, 42
modules, 14
modulus, 8
multiplicative
    identity, 10
    inverse, 10
multiplicatively
    closed set, 12
    invertible, 10

null matrix, 17

operation
    n-ary, 9
    binary, 9
output
    alphabet, 34
    function, 34

polynomial, 11
    constant, 11
    degree, 11
    linear, 11
    matrices, 19
    monic, 11
post-initial linear transducer, 113
Principal Ideal Domain, 12

quotient transducer, 40

relation, 7
restriction of a binary relation, 7
right module, 14
ring, 9
    commutative, 11
    of formal power series, 11
    of fractions, 13
    of polynomials, 10

scalar multiplication, 14
set of states, 34
size, 53
Smith normal form, 25
standard basis, 16, 24
state transition function, 34
structural
    matrices, 54
    parameters, 54
subspace, 15
symbols, 33

transfer function matrix, 85
trivial expansion, 73

unit, 10

vector space, 14
    dimension, 16
    finite dimensional, 16
    infinite dimensional, 16
    isomorphism, 28

weakly
    invertible, 45

invertible with delay, 45
word, 33
zero matrix, 17